Part No. 060755-00, Rev. A
January 2022

OmniSwitch AOS Release 8 Network Configuration Guide

8.8R1

This user guide covers multiple OmniSwitch product lines and describes overall AOS feature configuration information. For platform specific feature support, please refer to the Specifications Guide and the Release Notes.

Alcatel·Lucent Enterprise

www.al-enterprise.com

This user guide documents AOS Release 8.8R1.
The functionality described in this guide is subject to change without notice.

26801 West Agoura Road
Calabasas, CA 91301
(818) 880-3500 FAX (818) 880-3505

Service & Support Contact Information
North America: 800-995-2696
Latin America: 877-919-9526
EMEA : +800 00200100 (Toll Free) or +1(650)385-2193
Asia Pacific: +65 6240 8484
Web: https://myportal.al-enterprise.com
Email: ebg_global_supportcenter@al-enterprise.com

Contents

About This Guide xl
Supported Platforms xl
Who Should Read this Manual? xl
When Should I Read this Manual? xl
What is in this Manual? xli
What is Not in this Manual? xli
How is the Information Organized? xli
Documentation Roadmap xlii
Related Documentation xliv
Technical Support xlv

Chapter 1 **Configuring Ethernet Ports** 1-1
In This Chapter 1-1
Ethernet Port Defaults 1-2
Ethernet Ports Overview 1-3
Configuring Ethernet Port Parameters 1-3
Enabling and Disabling Autonegotiation 1-3
Configuring Crossover Settings 1-3
Setting Interface Line Speed 1-3
Configuring Duplex Mode 1-4
Setting Trap Port Link Messages 1-4
Resetting Statistics Counters 1-4
Enabling and Disabling Interfaces 1-4
Configuring a Port Alias 1-5
Configuring Maximum Frame Sizes 1-5
Configuring Digital Diagnostic Monitoring (DDM) 1-5
Configuring Flood Rate Limiting 1-6
Configuring Flow Control 1-7
Enabling and Disabling Enhanced Port Performance (EPP) 1-8
Configuring Energy Efficient Ethernet (802.3az) 1-10
Configuring Split-Mode 1-10
Configuring Beacon LED 1-10
Using TDR Cable Diagnostics 1-12
Initiating a TDR Cable Diagnostics Test 1-12
Displaying TDR Test Results 1-12
Clearing TDR Test Statistics 1-13
Interfaces Violation Recovery 1-14

Violation Shutdown and Recovery Methods 1-14
Interaction With Other Features 1-15
Configuring Interface Violation Recovery 1-16
Verifying the Interfaces Violation Recovery Configuration 1-17

Clearing Ethernet Port Violations 1-18

Link Monitoring 1-19
Monitoring Interface Errors 1-19
Monitoring Interface Flapping 1-19
Monitoring Window 1-20
Configuring the Wait-to-Restore Timer 1-20
Configuring the Wait-to-Shutdown Timer 1-21
Displaying Link Monitoring Information 1-22

Link Fault Propagation 1-23
Interaction With Interfaces Violation Recovery 1-23
Configuring Link Fault Propagation 1-24
LFP Application Example 1-25

IEEE 1588 Precision Time Protocol (PTP) 1-26
Enabling/Disabling PTP Time Stamping 1-26
Enabling/Disabling PTP Peer-to-Peer Transparent Clock 1-27

MAC Security Overview 1-28
How It Works? 1-28
Enabling/Disabling MACsec on an Interface 1-31
Verifying the MACsec Configuration 1-32

Chapter 2 Configuring UDLD 2-1

In This Chapter 2-1

UDLD Defaults 2-2

Quick Steps for Configuring UDLD 2-3

UDLD Overview 2-4
UDLD Operational Mode 2-4
Mechanisms to Detect Unidirectional Links 2-5

Configuring UDLD 2-6
Enabling and Disabling UDLD 2-6
Configuring the Operational Mode 2-7
Configuring the Probe-Timer 2-7
Configuring the Echo-Wait-Timer 2-7
Clearing UDLD Statistics 2-8

Verifying the UDLD Configuration 2-8

Chapter 3 Managing Source Learning 3-1

In This Chapter 3-1

Source Learning Defaults 3-2

MAC Address Table Overview 3-3

Using Static MAC Addresses 3-3

Configuring Static MAC Addresses 3-4
Using Static Multicast MAC Addresses 3-5
Configuring Static Multicast MAC Addresses 3-5
Configuring MAC Address Table Aging Time 3-7
Configuring the Source Learning Status 3-8
Increasing the MAC Address Table Size 3-9
Displaying Source Learning Information 3-10

Chapter 4 Configuring VLANs 4-1
In This Chapter 4-1
VLAN Defaults 4-2
Sample VLAN Configuration 4-3
VLAN Management Overview 4-4
Creating/Modifying VLANs 4-4
Adding/Removing a VLAN 4-5
Enabling/Disabling the VLAN Administrative Status 4-5
Modifying the VLAN Description 4-5
Assigning Ports to VLANs 4-6
Changing the Default VLAN Assignment for a Port 4-6
Using 802.1Q Tagging 4-7
Enabling/Disabling Spanning Tree for a VLAN 4-9
Enabling/Disabling Source Learning 4-9
Configuring VLAN IP Interfaces 4-10
Bridging VLANs Across Multiple Switches 4-11
Verifying the VLAN Configuration 4-13
Understanding Port Output Display 4-13
Using Private VLANs 4-15
Private VLAN Ports 4-16
Quick Steps for Configuring PVLANs 4-16
PVLAN Management Overview 4-17
Creating PVLANs 4-18
Creating Secondary VLANs 4-19
Assigning Ports to PVLANs 4-20
Protocol Configuration Requirements for PVLAN 4-22
Sample PVLAN Use Case 4-24
Verifying the PVLAN Configuration 4-25

Chapter 5 Configuring High Availability VLANs 5-1
In This Chapter 5-1
High Availability Default Values 5-2
Quick Steps for Creating High Availability VLANs 5-3
High Availability VLAN Overview 5-4

High Availability VLAN Operational Mode 5-4
Traffic Flows in High Availability VLAN 5-5

Configuring High Availability VLANs on a Switch 5-6
Creating and Deleting VLANs 5-6
Adding and Removing Server Cluster Ports 5-7
Assigning and Modifying Server Cluster Mode 5-7
Assigning and Removing MAC Addresses 5-8

Application Examples 5-9
Example 1: Layer 2 Server Cluster 5-9
Example 2: Layer 3 Server Cluster 5-11
Example 3: Layer 3 Server Cluster with IP Multicast Address to Cluster (IGMP) 5-13

Displaying High Availability VLAN Status 5-16

Chapter 6 Configuring Spanning Tree Parameters 6-1

In This Chapter 6-2

Spanning Tree Bridge Parameter Defaults 6-3

Spanning Tree Port Parameter Defaults 6-3

Multiple Spanning Tree (MST) Region Defaults 6-4

Spanning Tree Overview 6-5
How the Spanning Tree Topology is Calculated 6-5

MST General Overview 6-12
How MSTP Works 6-12
Comparing MSTP with STP and RSTP 6-15
What is a Multiple Spanning Tree Instance (MSTI) 6-15
What is a Multiple Spanning Tree Region 6-16
What is the Common Spanning Tree 6-17
What is the Internal Spanning Tree (IST) Instance 6-17
What is the Common and Internal Spanning Tree Instance 6-17
MST Configuration Overview 6-17
MST Interoperability and Migration 6-18

Spanning Tree Operating Modes 6-20
Using Flat Spanning Tree Mode 6-20
Using Per-VLAN Spanning Tree Mode 6-21
Using Per-VLAN Spanning Tree Mode with PVST+ 6-22
OmniSwitch PVST+ Interoperability 6-23

Using Spanning Tree Configuration Commands 6-26

Configuring STP Bridge Parameters 6-26
Selecting the Spantree Protocol 6-27
Configuring the Bridge Priority 6-28
Configuring the Bridge Hello Time 6-29
Configuring the Bridge Max-Age Time 6-29
Configuring the Forward Delay Time for the Switch 6-30
Enabling/Disabling the VLAN BPDU Switching Status 6-30

Configuring the Path Cost Mode 6-31
Using Automatic VLAN Containment 6-31

Configuring STP Port Parameters 6-33
Enabling/Disabling Spanning Tree on a Port 6-34
Enabling/Disabling Loop-guard 6-35
Configuring Port Priority 6-35
Configuring Port Path Cost 6-36
Configuring Port Mode 6-38
Configuring Port Connection Type 6-40
Configuring the Edge Port Status 6-41
Restricting Port Roles (Root Guard) 6-42
Restricting TCN Propagation 6-42
Limiting BPDU Transmission 6-42

Sample Spanning Tree Configuration 6-43
Example Network Overview 6-43
Example Network Configuration Steps 6-44

Sample MST Region Configuration 6-46

Sample MSTI Configuration 6-48

Verifying the Spanning Tree Configuration 6-51

Chapter 7 **Configuring Shortest Path Bridging** 7-1

In This Chapter 7-2

SPBM Parameter Defaults 7-3

SPBM Interface Defaults 7-3

SPBM Service Defaults 7-4

Shortest Path Bridging Overview 7-5
SPBM Shortest Path Trees 7-7
SPB Services 7-11
Sample SPBM Network Topology 7-12
SPBM Pseudo-Wire (E-LINE Transparent) Service 7-14
Remote Fault Propagation for SPBM Services 7-15
IP over SPBM 7-18
SPB Over Shared Ethernet 7-23
SPB In-Band Management 7-25

Interaction With Other Features 7-32
Backbone VLANs (VLAN Manager) 7-32
IP Multicast Switching 7-32
Link Aggregation 7-33
OAM 7-33
Quality of Service (QoS) 7-33
Universal Network Profiles (UNP) 7-34
UniDirectional Link Detection (UDLD) 7-35
VRF 7-35

Quick Steps for Configuring SPBM 7-36
Quick Steps for Configuring the SPBM Backbone 7-36
Quick Steps for Configuring SPB Services 7-37
Sample Command Configuration 7-37

Configuring SPBM 7-39
Configure the SPBM Backbone (ISIS-SPB) 7-39
Configure SPBM Services 7-39
SPB Configuration Guidelines 7-40
Configuring BVLANs 7-42
Configuring SPB Interfaces 7-44
Configuring Global ISIS-SPB Parameters 7-47
Creating an SPB Service 7-52
Configuring Service Access Points (SAPs) 7-55
Configuring an SPB Pseudo-Wire Service 7-63
Configuring Remote Fault Propagation for SPBM 7-66
Configuring IP over SPB 7-73
Configuring SPB Over Shared Ethernet 7-101

Verifying the SPB Backbone and Services 7-107
Verifying the ISIS-SPB Backbone Configuration 7-107
Verifying the SPB Service Configuration 7-108

Chapter 8 **Configuring Loopback Detection** 8-1

In This Chapter 8-1

LBD Defaults 8-2

Quick Steps for Configuring LBD 8-3

LBD Overview 8-4
Transmission Timer 8-4

Remote-origin LBD Overview 8-4

Interaction With Other Features 8-6
Spanning Tree Protocol 8-6
Link Aggregation 8-6

Configuring LBD 8-7
Enabling LBD 8-7
Enabling Remote-origin LBD 8-7
Configuring the LBD Transmission Timer 8-8
Viewing LBD Statistics 8-8
Recovering a Port from LBD Shutdown 8-8
Configuring Autorecovery-timer for LBD Shutdown Ports 8-8

LBD for Service Access Interface 8-8
Enabling LBD on Service-access Interface 8-9
LBD Packet Processing Mechanism for LBD Service Access Ports 8-9
Sample Scenarios 8-10

Verifying the LBD Configuration 8-12

Chapter 9 **Configuring Static Link Aggregation** 9-1

In This Chapter 9-1

Static Link Aggregation Default Values 9-2

Quick Steps for Configuring Static Link Aggregation 9-3

Static Link Aggregation Overview 9-5
- Static Link Aggregation Operation 9-5
- Relationship to Other Features 9-6

Configuring Static Link Aggregation Groups 9-6
- Configuring Mandatory Static Link Aggregate Parameters 9-6
- Creating and Deleting a Static Link Aggregate Group 9-7
- Adding and Deleting Ports in a Static Aggregate Group 9-7

Modifying Static Aggregation Group Parameters 9-9
- Modifying the Static Aggregate Group Name 9-9
- Modifying the Static Aggregate Group Administrative State 9-9

Application Example 9-10

Displaying Static Link Aggregation Configuration and Statistics 9-11

Chapter 10 **Configuring Dynamic Link Aggregation** 10-1

In This Chapter 10-1

Dynamic Link Aggregation Default Values 10-2

Quick Steps for Configuring Dynamic Link Aggregation 10-3

Dynamic Link Aggregation Overview 10-5
- Dynamic Link Aggregation Operation 10-5
- Relationship to Other Features 10-7

Configuring Dynamic Link Aggregate Groups 10-7
- Configuring Mandatory Dynamic Link Aggregate Parameters 10-8
- Creating and Deleting a Dynamic Aggregate Group 10-8
- Configuring Ports to Join and Removing Ports in a Dynamic Aggregate Group 10-9

Modifying Dynamic Link Aggregate Group Parameters 10-11
- Modifying Dynamic Aggregate Group Parameters 10-11
- Modifying Dynamic Link Aggregate Actor Port Parameters 10-16
- Modifying Dynamic Aggregate Partner Port Parameters 10-19

Application Examples 10-25
- Sample Network Overview 10-25
- Link Aggregation and Spanning Tree Example 10-26
- Link Aggregation and QoS Example 10-27

Displaying Dynamic Link Aggregation Configuration and Statistics 10-28

Chapter 11 **Configuring Dual-Home Links** 11-1

In This Chapter 11-1

Dual-Home Link Active-Active Defaults 11-2

Dual-Home Link Active-Active 11-3
DHL Active-Active Operation 11-3
DHL Configuration Guidelines 11-6
Configuring DHL Active-Active 11-6
Dual-Home Link Active-Active Example 11-8
Recommended DHL Active-Active Topology 11-10
Unsupported DHL Active-Active Topology (Network Loops) 11-11

Displaying the Dual-Home Link Configuration 11-12

Chapter 12 Configuring ERP 12-1

In This Chapter 12-1

ERP Defaults 12-2

ERP Overview 12-3
ERP Basic Operation 12-5
ERPv2 Basic Operation 12-7

Interaction With Other Features 12-9

Quick Steps for Configuring ERP with Standard VLANs 12-10

Quick Steps for Configuring ERP with VLAN Stacking 12-11

ERP Configuration Overview and Guidelines 12-12

Configuring an ERP Ring 12-13
Adding VLANs to Ring Ports 12-13
Configuring an RPL Port 12-14
Setting the Wait-to-Restore Timer 12-14
Setting the Guard Timer 12-14
Configuring ERP with VLAN Stacking NNIs 12-15
Clearing ERP Statistics 12-16

ERPv2 Configuration Overview and Guidelines 12-17
Major and Sub Ring Management 12-17
Sample Switch Configuration 12-18

Sample Ethernet Ring Protection Configuration 12-21
Example ERP Overview 12-21
Example ERP Configuration Steps 12-22

Sample ERPv2 Ring Configuration 12-23
Example ERPv2 Overview 12-23
Configuring Shared Link 12-24
Configuring Switches in Main Ring 12-25
Configuring Secondary RPL Node 12-25
Configuring Switch in Sub Ring 12-25

Verifying the ERP Configuration 12-26

Chapter 13 Configuring MRP 13-1

In This Chapter 13-1

MRP Overview 13-2
MRP Operation 13-2

Media Redundancy Manager (MRM) 13-3
Media Redundancy Client (MRC) 13-4
Redundancy Domain 13-4
Media Redundancy Automanager (MRA) 13-4

MRP Configuration Overview and Guidelines 13-6
Configuring an MRP Ring 13-6
Configuring MRP Ring Parameters 13-6

Verifying MRP Configuration 13-7

Chapter 14 Configuring MVRP 14-1

In This Chapter 14-1

MVRP Defaults 14-2

Quick Steps for Configuring MVRP 14-3

MRP Overview 14-4

MVRP Overview 14-4
How MVRP Works 14-4
Interaction With Other Features 14-6

Configuring MVRP 14-7
Enabling MVRP 14-7
Configuring the Maximum Number of VLANs 14-7
Configuring MVRP Registration 14-8
Configuring the MVRP Applicant Mode 14-9
Modifying MVRP Timers 14-10
Restricting VLAN Registration 14-11
Restricting Static VLAN Registration 14-11
Restricting VLAN Advertisement 14-12

Verifying the MVRP Configuration 14-13

Chapter 15 Configuring 802.1AB 15-1

In This Chapter 15-1

802.1AB Defaults Table 15-2

Quick Steps for Configuring 802.1AB 15-3

802.1AB Overview 15-4
Mandatory TLVs 15-4
Optional TLVs 15-4
LLDP-Media Endpoint Devices 15-5
LLDP Agent Operation 15-6
LLDPDU Transmission and Reception 15-6
Aging Time 15-6
LLDP Agent Security Mechanism 15-7

Configuring 802.1AB 15-8
Configuring LLDPDU Flow 15-8
Enabling and Disabling Notification 15-8
Enabling and Disabling Management TLV 15-9
Enabling and Disabling 802.1 TLV 15-9

Enabling and Disabling 802.3 TLV 15-9
Enabling and Disabling MED TLV 15-10
Enabling and Disabling Proprietary TLV 15-10
Enabling and Disabling Application Priority TLV 15-12
Setting the Transmit Interval 15-13
Setting the Transmit Hold Multiplier Value 15-13
Setting the Reinit Delay 15-13
Setting the Notification Interval 15-13
Application Example - LLDP MED 15-14

Verifying 802.1AB Configuration 15-15

Chapter 16 Configuring SIP Snooping 16-1

In This Chapter 16-1

SIP Snooping Defaults 16-2

Parameter Description and Values 16-3

Quick Steps for Configuring SIP Snooping 16-4

SIP Snooping Overview 16-5

Using SIP Snooping 16-6
Interoperability 16-7

SIP Snooping Configuration Guidelines 16-8
Configuring Edge Port 16-8
Configuring Trusted SIP Server 16-8
Configuring SIP Snooping TCP Ports 16-9
Configuring SIP Snooping UDP Ports 16-9
Configuring the SIP Control DSCP 16-9
Configuring SOS Calls 16-9
Configuring SOS Call DSCP 16-9
Configuring RTCP Thresholds 16-10
Configuring the Logging Threshold for the Number of Calls 16-10
Configuring Policy Rules for SIP Snooping 16-10
Unsupported Topologies 16-11

SIP Snooping Use Case 16-12

SIP Snooping Limitations 16-15

Verifying the SIP Snooping Configuration 16-16

Chapter 17 Configuring IP 17-1

In This Chapter 17-1

IP Defaults 17-3

Quick Steps for Configuring IP Forwarding 17-3

IP Overview 17-4
IP Protocols 17-4

IP Forwarding 17-6
Configuring an IP Interface 17-7
Configuring a Loopback0 Interface 17-9

Configuring an IP Managed Interface 17-10
Creating a Static Route or Recursive Static Route 17-11
Creating a Default Route 17-12
Configuring a Blackhole Route 17-12
Configuring an IP Routed Port 17-13
Configuring Address Resolution Protocol (ARP) 17-13
Configuring Gratuitous ARP 17-17

IP Configuration 17-19
Configuring the Router Primary Address 17-19
Configuring the Router ID 17-19
Configuring the Route Preference of a Router 17-19
Configuring the Time-to-Live (TTL) Value 17-20
Configuring Route Map Redistribution 17-20
IP-Directed Broadcasts 17-26
Denial of Service (DoS) Filtering 17-27
Enabling/Disabling IP Services 17-31

Managing IP 17-33
Internet Control Message Protocol (ICMP) 17-33
Using the Ping Command 17-35
Tracing an IP Route 17-36
Transmission Control Protocol (TCP) 17-36
Displaying UDP Information 17-37

Tunneling 17-37
Generic Routing Encapsulation 17-37
IP Encapsulation within IP 17-37
Tunneling Operation 17-38
Configuring a Tunnel Interface 17-39

Verifying the IP Configuration 17-40

VRF Route Leak 17-41
Quick Steps for Configuring VRF Route Leak 17-41
Configuring VRF Route Leak 17-42
Verifying VRF Route Leak Configuration 17-43

Chapter 18 Configuring Multiple VRF 18-1

In This Chapter 18-1

VRF Defaults 18-2

Quick Steps for Configuring Multiple VRF 18-2

Multiple VRF Overview 18-5
VRF Profiles 18-7
Using the VRF Command Line Interface 18-7
ASCII-File-Only Syntax 18-8
Management VRF 18-8

VRF Interaction With Other Features 18-10
AAA RADIUS/TACACS+/LDAP Servers 18-10
BGPv4 18-11
IP-IP and GRE Tunnels 18-11
IPv6 Routing Protocols 18-11

Management Applications (Telnet and SSH) 18-11
FTP 18-11
NTP 18-12
WebView 18-12
Syslog Server 18-12
Quality of Service (QoS) 18-12
SNMP 18-12
VLANs 18-13
UDP/DHCP Relay 18-13

Configuring VRF Instances 18-14
Configuring the VRF Profile 18-14
Selecting a VRF Instance 18-15
Assigning IP Interfaces to a VRF Instance 18-16
Configuring Routing Protocols for a Specific VRF Instance 18-16
Removing a VRF Instance 18-16

Verifying the VRF Configuration 18-17

Chapter 19 Configuring IPv6 19-1

In This Chapter 19-1

IPv6 Defaults 19-2

Quick Steps for Configuring IPv6 Routing 19-3

IPv6 Overview 19-4
IPv6 Addressing 19-5
Tunneling IPv6 over IPv4 19-9
Local Proxy Neighbor Discovery (LPND) 19-12
Router Advertisement (RA) Filtering 19-12
Neighbor Cache Limit 19-12
Neighbor Unreachability Detection (NUD) 19-12

Configuring an IPv6 Interface 19-13
Configuring a Unique Local IPv6 Unicast Address 19-14
Modifying an IPv6 Interface 19-14
Removing an IPv6 Interface 19-14
Configuring an IPv6 Routed Port 19-15

Assigning IPv6 Addresses 19-16
Removing an IPv6 Address 19-17

Configuring IPv6 Tunnel Interfaces 19-18

Creating an IPv6 Static Route 19-19

Configuring the Route Preference of a Router 19-21

Configuring Route Map Redistribution 19-22

VRF Route Leak 19-28
Quick Steps for Configuring VRF Route Leak 19-28
Configuring VRF Route Leak 19-29
Verifying VRF Route Leak Configuration 19-30

Configuring Local Proxy Neighbor Discovery 19-31

Configuring Neighbor Cache Limit 19-31
Configuring Neighbor Unreachability Detection 19-31
Configuring Router Advertisement Filtering 19-33
Reply or Ignore Echo Requests 19-34
ICMPv6 Error Message Rate Limiting 19-34
IPv6 EMP Interface 19-35
Configure IPv6 EMP Interface 19-35
Verifying the IPv6 Configuration 19-37

Chapter 20 Configuring IPsec 20-1
In This Chapter 20-1
IPsec Defaults 20-2
Quick Steps for Configuring an IPsec AH Policy 20-3
Quick Steps for Configuring an IPsec Discard Policy 20-4
IPsec Overview 20-5
Encapsulating Security Payload (ESP) 20-5
Authentication Header (AH) 20-6
IPsec on the OmniSwtich 20-7
Securing Traffic Using IPsec 20-8
Discarding Traffic using IPsec 20-9
Configuring IPsec on the OmniSwitch 20-10
Configuring an IPsec Master Key 20-10
Configuring an IPsec Policy 20-11
Configuring an IPsec SA 20-15
Enabling and Disabling Default Discard Policy 20-18
Additional Examples 20-20
Configuring ESP 20-20
Discarding RIPng Packets 20-22
Verifying the IPsec Configuration 20-23

Chapter 21 Configuring RIP 21-1
In This Chapter 21-1
RIP Defaults 21-2
Quick Steps for Configuring RIP Routing 21-3
RIP Overview 21-4
RIP Version 2 21-5
RIP Routing 21-6
Loading RIP 21-6
Enabling RIP 21-7
Creating a RIP Interface 21-7
Enabling a RIP Interface 21-7

RIP Options 21-9
Configuring the RIP Forced Hold-Down Interval 21-9
Configuring the RIP Update Interval 21-9
Configuring the RIP Invalid Timer 21-10
Configuring the RIP Garbage Timer 21-10
Configuring the RIP Hold-Down Timer 21-10
Reducing the Frequency of RIP Routing Updates 21-10
Enabling a RIP Host Route 21-11

Configuring Redistribution 21-12

RIP Security 21-18
Configuring Authentication Type 21-18
Configuring Passwords 21-18

Verifying the RIP Configuration 21-19

Chapter 22 Configuring BFD 22-1

In This Chapter 22-1

BFD Defaults 22-2

Quick Steps for Configuring BFD 22-4
Quick Steps for Configuring BFD Support for Layer 3 Protocols 22-6

BFD Overview 22-10
Benefits of Using BFD For Failure Detection 22-10
How the BFD Protocol Works 22-10
Operational Mode and Echo Function 22-11
BFD Packet Formats 22-11
BFD Session Establishment 22-12

Configuring BFD 22-14
Configuring BFD Session Parameters 22-14
Configuring BFD Support for Layer 3 Protocols 22-18

BFD Application Example 22-33
Example Network Overview 22-33

Verifying the BFD Configuration 22-39

Chapter 23 Configuring DHCP Relay 23-1

In This Chapter 23-1

DHCP Relay Defaults 23-2

Quick Steps for Setting Up DHCP Relay 23-3

DHCP Relay Overview 23-4
DHCP 23-5
DHCP and the OmniSwitch 23-5
External DHCP Relay Application 23-6
Internal DHCP Relay 23-7

Configuring DHCP Relay 23-8
Configuring the Status of the DHCP Relay Feature 23-8
Setting the DHCP Relay Forwarding Mode 23-8

Configuring DHCP Relay Parameters 23-9
Configuring DHCP Relay for the SPB Service Domain 23-10

Configuring the DHCP Client Interface 23-13
Enabling the DHCP Client Interface 23-13
DHCP Server Preference in DHCP Client Interface 23-14
Configuring DHCP Server Preference 23-15

Configuring Generic UDP Relay 23-16
Configuration Guidelines 23-16
Enabling/Disabling Generic UDP Relay 23-17
Specifying a Forwarding VLAN 23-17
Specifying a Forwarding SPB Service 23-18
Specifying a Forwarding IP Address 23-19

Configuring DHCP Security Features 23-20
Using the Relay Agent Information Option (Option-82) 23-20
Using DHCP Snooping 23-23

Verifying the DHCP Relay Configuration 23-30

DHCPv6 Relay Overview 23-31
Quick Steps for Configuring DHCPv6 Relay 23-31
DHCPv6 Relay Interface 23-31
DHCPv6 Relay Messages 23-32

Configuring DHCPv6 Relay 23-32
Enabling the DHCPv6 Relay Service 23-32
Configuring the DHCPv6 Relay Interface 23-32
Setting Maximum Hops 23-33
Verifying the DHCPv6 Relay Configuration 23-33

Using DHCPv6 Snooping 23-34
Enabling DHCPv6 Snooping 23-34
Configuring the DHCPv6 Snooping Binding Table 23-34
Configuring IPv6 Source Filtering (ISF) 23-36
Using IPv6 DHCP Guard 23-38

Verifying the DHCPv6 Configuration 23-40

Chapter 24 Configuring an Internal DHCP Server 24-1

In This Chapter 24-1

DHCP Server Default Values 24-2

Quick Steps to Configure Internal DHCP Server 24-2

DHCP Server Overview 24-4
The DHCP process 24-4
Internal DHCP Server on OmniSwitch 24-4
VitalQIP Server 24-4

Interaction With Other Features 24-5
Virtual Router Forwarding (VRF) 24-5
DHCP Snooping 24-5
IP Interfaces 24-5
VitalQIP Server 24-5

Configuring DHCP Server on OmniSwitch 24-6
Policy file 24-6
DHCP Configuration Files 24-7
DHCP Server Database file 24-10

DHCP Server Application Example 24-11

Verifying the DHCP Server Configuration 24-13

Configuration File Parameters and Syntax 24-14

Policy File Parameters and Syntax 24-25

Chapter 25 **Configuring VRRP** 25-1

In This Chapter 25-1

VRRP Defaults 25-2

Quick Steps for Creating a Virtual Router 25-4

VRRP Overview 25-5
Why Use VRRP? 25-6
Definition of a Virtual Router 25-6
VRRP MAC Addresses 25-7
VRRP Startup Delay 25-8
VRRP Tracking 25-8
Configuring Collective Management Functionality 25-8

Interaction With Other Features 25-8
IPv4 and IPv6 Interfaces 25-8
VRRP over UNP Dynamic SPB SAPs 25-9
VRRP Tracking with BFD 25-9
Virtual Routing and Forwarding (VRF) 25-10

VRRP Configuration Overview 25-11
Basic Virtual Router Configuration 25-11
Creating/Deleting a Virtual Router 25-11
Specifying an IP Address for a Virtual Router 25-13
Configuring the Advertisement Interval 25-14
Configuring Virtual Router Priority 25-15
Setting Preemption for Virtual Routers 25-16
Setting the Accept Mode 25-16
Configuring the VRRP Version 25-17
Enabling/Disabling a Virtual Router 25-17
Setting VRRP Traps 25-18
Setting VRRP Startup Delay 25-18
Configuring Collective Management Functionality 25-19

Creating VRRP Tracking Policies 25-23
Associating a Tracking Policy with a Virtual Router 25-23

Verifying the VRRP Configuration 25-25

IPv4 VRRP Application Example 25-26
IPv4 VRRP Tracking Example 25-27

IPv6 VRRP Application Example 25-29
IPv6 VRRP Tracking Example 25-30

Chapter 26 **Configuring Server Load Balancing** 26-1

In This Chapter 26-1

Server Load Balancing Default Values 26-2

Quick Steps for Configuring Server Load Balancing 26-3
Quick Steps for Configuring a QoS Policy Condition Cluster 26-4

Server Load Balancing Overview 26-6
Server Load Balancing Cluster Identification 26-6
Server Load Balancing Example 26-7
Weighted Round Robin Distribution Algorithm 26-8
Server Health Monitoring 26-9

Configuring Server Load Balancing on a Switch 26-10
Enabling and Disabling Server Load Balancing 26-10
Configuring and Deleting SLB Clusters 26-11
Assigning Servers to and Removing Servers from a Cluster 26-13

Modifying Optional Parameters 26-14
Modifying the Ping Period 26-14
Modifying the Ping Timeout 26-14
Modifying the Ping Retries 26-15
Modifying the Relative Weight of a Physical Server 26-15

Taking Clusters and Servers On/Off Line 26-16
Taking a Cluster On/Off Line 26-16
Taking a Server On/Off Line 26-16

Configuring SLB Probes 26-17
Creating SLB Probes 26-17
Deleting SLB Probes 26-17
Associating a Probe with a Cluster 26-18
Associating a Probe with a Server 26-18
Modifying SLB Probes 26-18

Displaying Server Load Balancing Status and Statistics 26-21

Chapter 27 **Configuring IP Multicast Switching** 27-1

In This Chapter 27-1

IPMS Default Values 27-2

IPMSv6 Default Values 27-3

IPMS Overview 27-4
IPMS Example 27-4
Reserved IP Multicast Addresses 27-5
IP Multicast Routing 27-5

Interaction With Other Features 27-7
IPMS for Shortest Path Bridging 27-7
VLAN and Service Domains 27-7

Configuring IPMS on a Switch 27-10
Enabling and Disabling IP Multicast Status 27-10
Enabling and Disabling Flooding of Unknown Multicast Traffic 27-11

Enabling and Disabling IGMP Querier-forwarding 27-12
Configuring and Restoring the IGMP Version 27-13
Configuring and Removing an IGMP Static Neighbor 27-13
Configuring and Removing an IGMP Static Querier 27-14
Configuring and Removing an IGMP Static Group 27-15
Initial Multicast Packet Buffering 27-16

Modifying IPMS Parameters 27-18
Modifying the IGMP Query Interval 27-18
Modifying the IGMP Last Member Query Interval 27-19
Modifying the IGMP Query Response Interval 27-20
Enabling and Disabling Zero-based IGMP Query 27-20
Modifying the IGMP Router Timeout 27-21
Modifying the Source Timeout 27-22
Enabling and Disabling IGMP Querying 27-23
Modifying the IGMP Robustness Variable 27-24
Enabling and Disabling the IGMP Spoofing 27-25
Enabling and Disabling the IGMP Zapping 27-26
Limiting IGMP Multicast Groups 27-26

IPMSv6 Overview 27-28
IPMSv6 Example 27-28
Reserved IPv6 Multicast Addresses 27-29
MLD Version 2 27-29

Configuring IPMSv6 on a Switch 27-30
Enabling and Disabling IPv6 Multicast Status 27-30
Enabling and Disabling Flooding of Unknown Multicast Traffic 27-31
Enabling and Disabling MLD Querier-forwarding 27-32
Configuring and Restoring the MLD Version 27-32
Configuring and Removing an MLD Static Neighbor 27-33
Configuring and Removing an MLD Static Querier 27-34
Configuring and Removing an MLD Static Group 27-35

Modifying IPMSv6 Parameters 27-36
Modifying the MLD Query Interval 27-36
Modifying the MLD Last Member Query Interval 27-37
Modifying the MLD Query Response Interval 27-37
Enabling and Disabling Zero-based MLD Query 27-38
Modifying the MLD Router Timeout 27-39
Modifying the Source Timeout 27-40
Enabling and Disabling the MLD Querying 27-40
Modifying the MLD Robustness Variable 27-41
Enabling and Disabling MLD Spoofing 27-42
Enabling and Disabling the MLD Zapping 27-43
Limiting MLD Multicast Groups 27-44

IPMS Application Example 27-45

IPMSv6 Application Example 27-47

Displaying IPMS Configurations and Statistics 27-49

Displaying IPMSv6 Configurations and Statistics 27-50

Chapter 28 **Configuring QoS** 28-1

In This Chapter 28-2

QoS General Overview 28-3

Classification 28-5
- How Traffic is Classified and Marked 28-5
- Configuring Trusted Ports 28-9

Congestion Management 28-11
- Queue Sets 28-11
- QSet Profiles 28-13
- Multicast and Unicast Traffic Distribution 28-17
- Multicast Source PFC on the OmniSwitch 6900 28-19

OmniSwitch Congestion Avoidance 28-20

Traffic Policing and Shaping 28-21
- Policing 28-21
- Shaping 28-21
- Tri-Color Marking 28-22
- Configuring Policy Bandwidth Policing 28-25
- Configuring Port Bandwidth Policing 28-27

QoS Policy Overview 28-29
- How Policies Are Used 28-29
- Policy Lists 28-30
- Interaction With Other Features 28-30
- Valid Policies 28-30
- Policy Conditions 28-31
- Policy Actions 28-32

QoS Defaults 28-34
- Global QoS Defaults 28-34
- QoS Port Defaults 28-34
- Queue Management Defaults 28-35
- Policy Rule Defaults 28-36
- Policy Action Defaults 28-36
- Default (Built-in) Policies 28-37

Configuring QoS 28-38

Configuring Global QoS Parameters 28-39
- Enabling/Disabling QoS 28-39
- Using the QoS Log 28-39
- Setting the Statistics Interval 28-42
- Returning the Global Configuration to Defaults 28-42
- Verifying Global Settings 28-42

Creating Policies 28-43
- Quick Steps for Creating Policies 28-43
- ASCII-File-Only Syntax 28-44
- Creating Policy Conditions 28-45
- Creating Policy Actions 28-46
- Creating Policy Rules 28-47

Creating Policy Lists 28-50
Verifying Policy Configuration 28-53

Using Condition Groups in Policies 28-54
Sample Group Configuration 28-54
Creating Network Groups 28-55
Creating Services 28-56
Creating Service Groups 28-57
Creating MAC Groups 28-58
Creating Port Groups 28-59
Verifying Condition Group Configuration 28-60

Using Map Groups 28-61
Sample Map Group Configuration 28-61
How Map Groups Work 28-62
Creating Map Groups 28-62
Verifying Map Group Configuration 28-63

Using Access Control Lists 28-64
Layer 2 ACLs 28-64
Layer 3 ACLs 28-66
IPv6 ACLs 28-67
Multicast Filtering ACLs 28-68
Using ACL Security Features 28-68

Applying the Configuration 28-72
Interaction With LDAP Policies 28-73
Verifying the Applied Policy Configuration 28-74

Policy Applications 28-75
Basic QoS Policies 28-76
Redirection Policies 28-77
Policy Based Mirroring 28-78
ICMP Policy Example 28-79
802.1p and ToS/DSCP Marking and Mapping 28-79
Policy Based Routing 28-80

Enhanced Congestion Notification (ECN) 28-84
ECN Mechanism 28-84
Enhanced Congestion Notification (ECN) Profile Configuration 28-85

Chapter 29 Managing Policy Servers 29-1

In This Chapter 29-1

Policy Server Defaults 29-2

Policy Server Overview 29-3

Installing the LDAP Policy Server 29-3

Modifying Policy Servers 29-4
Modifying LDAP Policy Server Parameters 29-4
Disabling the Policy Server From Downloading Policies 29-4
Modifying the Port Number 29-5
Modifying the Policy Server Username and Password 29-5
Modifying the Searchbase 29-5

Configuring a Secure Socket Layer for a Policy Server 29-6
Loading Policies From an LDAP Server 29-6
Removing LDAP Policies From the Switch 29-6
Interaction With CLI Policies 29-7

Verifying the Policy Server Configuration 29-7

Chapter 30 Configuring Access Guardian 30-1

In This Chapter 30-2

Access Guardian Defaults 30-3
Access Guardian Global Configuration Defaults 30-3
Access Guardian Profile Defaults 30-4
Access Guardian UNP Port Defaults 30-5
Access Guardian Global AAA Parameter Defaults 30-6
Access Guardian AAA Profile Defaults 30-7
Access Guardian Captive Portal Defaults 30-8
Access Guardian Captive Portal Profile Defaults 30-9
Access Guardian QMR Defaults 30-9

Quick Steps for Configuring Access Guardian 30-10

Access Guardian Overview 30-12
Device Authentication 30-13
Device Classification 30-14
Role-based Access 30-15
UNP Profiles 30-16
UNP Ports 30-22
UNP Classification Rules 30-24
How it Works 30-26

Interaction With Other Features 30-27
Authentication, Authorization, and Accounting (AAA) 30-27
Bring Your Own Devices (BYOD) 30-27
Learned Port Security 30-27
Multiple VLAN Registration Protocol (MVRP) 30-28
Quality of Service (QoS) 30-29
Service Assurance Agent 30-29
Service Manager 30-29
Source Learning 30-30
Universal Network Profile (UNP) 30-30
UNP Dynamic SAPs 30-31

Configuring Port-Based Network Access Control 30-33
Setting Authentication Parameters for the Switch 30-34
Configuring UNP Port-Based Functionality 30-41
Configuring UNP Profiles 30-58
Configuring the UNP Profile Mapping 30-62
Setting the RADIUS Server Attribute Precedence 30-72
Configuring System Default Profile Parameters 30-74
Configuring QoS Policy Lists 30-76
Configuring UNP Classification Rules 30-80

Using Router Domain Authentication 30-84
Configuration Overview and Guidelines 30-84

Configuring Router Authentication 30-86
Router Domain Authentication Example 30-89

Using Captive Portal Authentication 30-91
Configuration Tasks and Guidelines 30-92
Quick Steps for Configuring Captive Portal Authentication 30-93
Configuring the Captive Portal Operating Mode 30-94
Using Captive Portal Configuration Profiles 30-95
Replacing the Captive Portal Certificate 30-96
Customizing Captive Portal Web Pages 30-97
Authenticating with Captive Portal 30-98

OmniAccess Stellar AP Integration 30-101
How it Works 30-102
AP Mode Configuration Guidelines - VLAN Domain 30-104
AP Mode Configuration Guidelines - SPB Service Domain 30-106
OmniAccess Stellar AP Configuration Guidelines 30-107
Quick Steps for Configuring OmniSwitch AP Discovery 30-108
Verify the OmniSwitch Configuration 30-110

Using L2 GRE Tunneling 30-113
Configuration Overview and Guidelines 30-113
Quick Steps for Configuring L2 GRE Tunneling 30-119
L2 GRE Tunneling Configuration Example 30-123

Using Quarantine Manager and Remediation 30-128

Access Guardian Application Examples 30-130
Application Example 1: Classification (Port Mobility) 30-131
Application Example 2: 802.1X Authentication 30-132
Application Example 3: Internal Captive Portal Authentication 30-134
Application Example 4: Supplicant/Non-supplicant with Captive Portal Authentication 30-136
Application Example 5: IP Phone (LLDP Network Policy TLV/ Mobile Tag) 30-139
Application Example 6: Restricted Role (Policy List) Assignment 30-141

Verifying Access Guardian Users 30-144
Logging Users Out of the Network 30-147

Verifying the Access Guardian Configuration 30-149

Bring Your Own Devices (BYOD) Overview 30-150
Key Components of a BYOD Solution 30-151
Configuring OmniSwitch BYOD Support 30-158
BYOD Authentication Process Overview 30-161
Multicast Domain Name System 30-162
Simple Service Discovery Protocol 30-163
Zero Configuration Networking (mDNS and SSDP) 30-167

BYOD Application Examples 30-180
Application Example 1: 802.1X — OmniSwitch Configuration 30-181
Application Example 1: 802.1X — ClearPass Configuration 30-182
Application Example 2: IP Phone — OmniSwitch Configuration 30-187
Application Example 2: IP Phone — ClearPass Configuration 30-188
Application Example 3: Guest — OmniSwitch Configuration 30-191

Application Example 3: Guest — ClearPass Configuration 30-192
Verifying the BYOD Configuration 30-196

IoT Device Profiling 30-197
IoT Device Profiling Overview 30-198
Quick Steps for Configuring Device Profile 30-199
Device Identification 30-199
Local Database and Signature Management 30-200
Automatic UNP Profile Assignment in Device Profiling 30-201
UNP Enforcement of Device Profile 30-202
Verifying the Device Profile Configuration 30-203

Chapter 31 Configuring Application Monitoring and Enforcement 31-1

In This Chapter 31-2

AppMon Defaults 31-3

Application Monitoring and Enforcement Overview 31-4
Application Monitoring 31-4
Application Enforcement 31-5
Quick Steps for Configuring AppMon 31-7

Application Signature File/Kit 31-8
Signature File Update 31-8
Application Flow Database 31-9

Configuring AppMon 31-10
Configuration Guidelines 31-11
Enabling/Disabling AppMon 31-13
Enabling/Disabling AppMon Per Port or Slot 31-13
Create Auto-Groups 31-14
Configuring Application Group 31-14
Configuring Application List 31-15
Activate Applications for AppMon 31-15
Configuring L3 Mode of Operation 31-16
Configuring L4 Mode of Operation 31-16
Configuring Enforcement Mode for IPv4/IPv6 Flows 31-16
Clearing Flow Table Entries 31-17
Configuring Flow Table Statistics Update 31-17
Configuring Aging Interval 31-17
Configuring Logging Threshold 31-18
Configuring Sync Interval 31-18
Configuring Force Flow Sync 31-18
Clearing Application List 31-19
Configuring AppMon Enforcement QoS Policy Rules 31-19
Separate File for AppMon Configuration 31-20

Verifying AppMon Configuration 31-21

Chapter 32 Configuring Application Fingerprinting 32-1

In This Chapter 32-1

AFP Defaults 32-2
Default REGEX Application Signatures 32-2

Quick Steps for Configuring AFP 32-4

AFP Overview 32-5
Application Fingerprinting Modes 32-6
Using the Application REGEX Signature File 32-7
Application Fingerprinting Database 32-8

Interaction With Other Features 32-9
General 32-9
QoS 32-9
sFLOW 32-9

Configuring AFP 32-10
Configuration Guidelines 32-10
Enabling/Disabling AFP 32-11
Enabling/Disabling Trap Generation 32-11
Changing the REGEX Signature Filename 32-12
Reloading the REGEX Signature File 32-12
Defining Application REGEX Signatures and Groups 32-13
Configuring AFP Port Modes 32-16

Verifying the AFP Configuration 32-18

Chapter 33 Managing Authentication Servers 33-1

In This Chapter 33-1

Server Defaults 33-2
RADIUS Authentication Servers 33-2
TACACS+ Authentication Servers 33-2
LDAP Authentication Servers 33-2

Quick Steps For Configuring Authentication Servers 33-4

Server Overview 33-5
Backup Authentication Servers 33-5
Authenticated Switch Access 33-5

RADIUS Servers 33-7
RADIUS Server Attributes 33-7
Configuring the RADIUS Client 33-13
RADIUS over TLS 33-14
RADIUS Health Check 33-14
RADIUS Server Statistics 33-15
Setting UNP Profile Precedence 33-19

TACACS+ Server 33-20
TACACS+ Client Limitations 33-20
Configuring the TACACS+ Client 33-21

LDAP Servers 33-22
Setting Up the LDAP Authentication Server 33-22
LDAP Server Details 33-23
Directory Server Schema for LDAP Authentication 33-28
Configuring the LDAP Authentication Client 33-32
Configuring OpenSSL Ciphers 33-34
Configuring Public Key Infrastructure (PKI) 33-36

Verifying the Authentication Server Configuration 33-38

Kerberos Snooping Overview 33-39
Importance of Kerberos Authentication 33-39
Kerberos Snooping Authentication 33-40
Configuring Kerberos Snooping 33-41
Verifying Kerberos Snooping Configuration 33-43

Chapter 34 Configuring Port Mapping 34-1

In This Chapter 34-1

Port Mapping Defaults 34-2

Quick Steps for Configuring Port Mapping 34-3

Creating/Deleting a Port Mapping Session 34-4
Creating a Port Mapping Session 34-4
Deleting a Port Mapping Session 34-4

Enabling/Disabling a Port Mapping Session 34-5
Enabling a Port Mapping Session 34-5
Disabling a Port Mapping Session 34-5
Disabling the Flooding of Unknown Unicast Traffic 34-5

Configuring a Port Mapping Direction 34-5
Configuring Unidirectional Port Mapping 34-5
Restoring Bidirectional Port Mapping 34-5

Sample Port Mapping Configuration 34-6
Example Port Mapping Overview 34-6
Example Port Mapping Configuration Steps 34-7

Verifying the Port Mapping Configuration 34-7

Chapter 35 Configuring Learned Port Security 35-1

In This Chapter 35-1

Learned Port Security Defaults 35-2

Sample Learned Port Security Configuration 35-3

Learned Port Security Overview 35-5
LPS Learning Window 35-5
MAC Address Types 35-6
How LPS Authorizes Source MAC Addresses 35-6
Dynamic Configuration of Authorized MAC Addresses 35-8
Static Configuration of Authorized MAC Addresses 35-8
Understanding the LPS Table 35-9

Interaction With Other Features 35-9
Access Guardian 35-10
Universal Network Profile (UNP) 35-10

Configuring Learned Port Security 35-11
Configuring the LPS Port Administrative Status 35-11
Configuring the LPS Learning Window 35-12
Configuring Learning Window Parameters 35-13

Configuring the Number of Bridged MAC Addresses Allowed 35-16
Configuring the Number of Filtered MAC Addresses Allowed 35-17
Configuring an Authorized MAC Address Range 35-17
Selecting the Security Violation Mode 35-19

Displaying Learned Port Security Information 35-20

Chapter 36 Diagnosing Switch Problems 36-1

In This Chapter 36-1

Port Mirroring Overview 36-3
Port Mirroring Defaults 36-3
Quick Steps for Configuring Port Mirroring 36-3

Port Monitoring Overview 36-4
Port Monitoring Defaults 36-4
Quick Steps for Configuring Port Monitoring 36-4

sFlow Overview 36-5
sFlow Defaults 36-5
Quick Steps for Configuring sFlow 36-5

Remote Monitoring (RMON) Overview 36-7
RMON Probe Defaults 36-7
Quick Steps for Enabling/Disabling RMON Probes 36-7

Switch Health Overview 36-8
Switch Health Defaults 36-8
Quick Steps for Configuring Switch Health 36-8

Port Mirroring 36-9
What Ports Can Be Mirrored? 36-9
How Port Mirroring Works 36-9
What Happens to the Mirroring Port 36-10
Mirroring on Multiple Ports 36-10
Using Port Mirroring with External RMON Probes 36-10
Remote Port Mirroring 36-12
Creating a Mirroring Session 36-13
Policy Based Multiple Destination Mirroring 36-14
Unblocking Ports (Protection from Spanning Tree) 36-14
Enabling or Disabling Mirroring Status 36-14
Disabling a Mirroring Session (Disabling Mirroring Status) 36-14
Configuring Port Mirroring Direction 36-15
Destination Tag-remove 36-15
Enabling or Disabling a Port Mirroring Session (Shorthand) 36-16
Displaying Port Mirroring Status 36-16
Deleting A Mirroring Session 36-16
Configuring Remote Port Mirroring 36-17
Configuring Policy Based Multiple Destination Mirroring 36-18
Configuring Remote Port Mirroring (RPMIR) Loopback Session for Multiple Destination Ports 36-19

Port Monitoring 36-22
Configuring a Port Monitoring Session 36-22
Enabling a Port Monitoring Session 36-23

Disabling a Port Monitoring Session 36-23
Deleting a Port Monitoring Session 36-23
Pausing a Port Monitoring Session 36-24
Configuring Port Monitoring Session Persistence 36-24
Configuring a Port Monitoring Data File 36-24
Configuring Port Monitoring Direction 36-25
Configuring the Capture Type 36-26
Displaying Port Monitoring Status and Data 36-26

sFlow 36-27
sFlow Manager 36-27
Receiver 36-27
Sampler 36-28
Poller 36-28
Configuring a sFlow Session 36-28
Configuring a Fixed Primary Address 36-29
Displaying a sFlow Receiver 36-29
Displaying a sFlow Sampler 36-30
Displaying a sFlow Poller 36-30
Displaying a sFlow Agent 36-30
Deleting a sFlow Session 36-31

Remote Monitoring (RMON) 36-32
Enabling or Disabling RMON Probes 36-34
Displaying RMON Tables 36-35

Monitoring Switch Health 36-39
Configuring Resource Thresholds 36-40
Displaying Health Threshold Limits 36-41
Configuring Sampling Intervals 36-42
Viewing Sampling Intervals 36-42
Viewing Health Statistics for the Switch 36-43
Viewing Health Statistics for a Specific Interface 36-44

Chapter 37 Configuring VLAN Stacking 37-1

In This Chapter 37-1

VLAN Stacking Defaults 37-2

VLAN Stacking Overview 37-3
How VLAN Stacking Works 37-5
VLAN Stacking Services 37-6

Interaction With Other Features 37-7

Quick Steps for Configuring VLAN Stacking 37-8

Configuring VLAN Stacking Services 37-10
Configuring SVLANs 37-11
Configuring a VLAN Stacking Service 37-12
Configuring VLAN Stacking Network Ports 37-13
Configuring a VLAN Stacking Service Access Point 37-15
Configuring VLAN Stacking User Ports 37-16
Configuring the Type of Customer Traffic to Tunnel 37-16
Configuring a Service Access Point Profile 37-18

Configuring a UNI Profile 37-20
Transparent Bridging 37-26

VLAN Stacking Application Example 37-27
VLAN Stacking Configuration Example 37-28

Wire-Rate Hardware Loopback Test 37-30
Configuring an Ethernet Loopback Test 37-30

Verifying the VLAN Stacking Configuration 37-35

Chapter 38 Using Switch Logging 38-1

In This Chapter 38-1

Switch Logging Defaults 38-2

Quick Steps for Configuring Switch Logging 38-2

Switch Logging Overview 38-3

Switch Logging Commands Overview 38-4
Enabling Switch Logging 38-4
Setting the Switch Logging Severity Level 38-4
Specifying the Switch Logging Output Device 38-5
Configuring the Switch Logging File Size 38-7
Clearing the Switch Logging Files 38-7
Displaying Switch Logging Records 38-8
Readable Customer Event Logs 38-8
Specifying the Switch Logging Format 38-10
Switch Logging Notifications 38-10
Specifying the Switch Logging Record Storage Limit 38-10

Chapter 39 Configuring Ethernet OAM 39-1

In This Chapter 39-1

Ethernet OAM Defaults 39-2

Ethernet OAM Overview 39-3
Ethernet Service OAM 39-3

Quick Steps for Configuring Ethernet OAM 39-8

Configuring Ethernet OAM 39-9
Configuring a Maintenance Domain 39-9
Configuring a Maintenance Association 39-10
Configuring a Maintenance End Point 39-11
Configuring a Virtual Maintenance End Point 39-11
Configuring Loopback 39-12
Configuring Linktrace 39-12
Configuring the Fault Alarm Time 39-12
Configuring the Fault Reset Time 39-12
Configuring Ethernet Frame Delay Measurement 39-12

Verifying the Ethernet OAM Configuration 39-14

Chapter 40 **Configuring EFM (LINK OAM)** 40-1

In This Chapter 40-1

LINK OAM Defaults 40-2

Quick Steps for Configuring LINK OAM 40-3

LINK OAM Overview 40-4
- Discovery 40-5
- Link Monitoring 40-5
- Remote Fault detection 40-5
- Remote Loopback Testing 40-6

Interaction With Other Features 40-6

Configuring LINK OAM 40-7
- Enabling and Disabling LINK OAM 40-7
- Setting the Transmit Delay 40-7
- Enabling and Disabling Propagation of Events 40-8

Configuring Link Monitoring 40-8
- Enabling and Disabling Errored frame period 40-8
- Enabling and Disabling Errored frame 40-8
- Enabling and Disabling Errored frame seconds summary 40-9

Configuring LINK OAM Loopback 40-9
- Enabling and Disabling Remote loopback 40-9

Verifying the LINK OAM Configuration 40-10

Chapter 41 **Configuring CPE Test Head** 41-1

In This Chapter 41-2

Quick Steps for Configuring CPE Test Head 41-3

CPE Test Head Overview 41-5

CPE Test Head Configuration Overview 41-6
- Configuration Guidelines 41-6

Configuring a CPE Test Profile 41-7

Configuring the L2 SAA Test 41-9

Running a CPE Test 41-10
- Stopping the CPE Test 41-10

Verifying the CPE Test Configuration and Results 41-11

Configuring CPE Test Group 41-13

Quick Steps for Configuring CPE Test Group 41-14

CPE Test Group Overview 41-17

CPE Test Group Configuration Overview 41-18
- Configuration Guidelines 41-19

Configuring a CPE Test Group Profile 41-20

Running a CPE Test Group test 41-22
- Stopping the CPE Test Group test 41-22

Verifying the CPE Test Group Configuration and Results 41-23

CPE Test Head Advanced Configuration 41-25
Running L2 SAA test 41-25
Configuring Remote Sys MAC 41-25
Saving the test results on the /flash 41-26

Sample Test Configurations 41-27
Sample Unidirectional Test Configuration 41-27
Sample Bidirectional Test Configuration 41-28
Sample Bidirectional Multi-stream Test Configuration 41-29

Chapter 42 Configuring PPPoE Intermediate Agent 42-1

In This Chapter 42-1

PPPoE-IA Defaults 42-1

Quick Steps for Configuring PPPoE-IA 42-2

PPPoE Intermediate Agent Overview 42-4
How PPPoE-IA Works 42-4

Configuring PPPoE-IA 42-5
Enabling PPPoE-IA Globally 42-5
Enabling PPPoE-IA on a Port 42-5
Configuring a Port as Trust or Client 42-5
Configuring Access Node Identifier for PPPoE-IA 42-6
Configuring Circuit Identifier 42-6
Configuring Remote Identifier 42-7

Verifying PPPoE-IA Configuration 42-8

Chapter 43 Configuring Service Assurance Agent 43-1

In This Chapter 43-1

SAA Defaults 43-2

Quick Steps for Configuring SAA 43-3

Service Assurance Agent Overview 43-4

Configuring Service Assurance Agent 43-5
Configuring an SAA ID 43-5
Configuring SAA SPB Session Parameters 43-7
Generating an SAA XML History File 43-8

Verifying the SAA Configuration 43-11

Appendix A Software License and Copyright Statements A-1

ALE USA, Inc. License Agreement A-1
ALE USA, INC. SOFTWARE LICENSE AGREEMENT A-1

Third Party Licenses and Notices A-4

Index Index-1

List of Figures

Figure 1-1 : Link Fault Propagation - Application Example. 1-25
Figure 1-2 : MAC Security Overview. 1-28
Figure 4-1 : Tagged and Untagged Traffic Network. 4-7
Figure 4-2 : VLAN Bridging Domain: Physical Configuration. 4-11
Figure 4-3 : VLAN Bridging Domain: Logical View. 4-12
Figure 4-4 : Using Private VLANs. 4-15
Figure 4-5 : PVLAN Spanning across Multiple Systems. 4-24
Figure 5-1 : Example of an L2 Server Cluster - Ingress to Egress Port Flow. 5-5
Figure 5-2 : Switch connected to an L2 Server Cluster through 3 ports (1/3, 1/4, 1/5). 5-9
Figure 5-3 : Switch connected to an L3 Server Cluster through 3 ports (1/3,1/4,1/5). 5-11
Figure 5-4 : Switch connected to an L3 Server Cluster (IGMP) through 3 ports (1/3,1/4,1/5). 5-13
Figure 6-1 : Physical Topology Example. 6-10
Figure 6-2 : Active Spanning Tree Topology Example. 6-11
Figure 6-3 : Per-VLAN Mode STP/RSTP. 6-13
Figure 6-4 : Flat Mode STP/RSTP (802.1D/802.1w). 6-13
Figure 6-5 : Flat Mode MSTP.. 6-14
Figure 6-6 : Multiple Spanning Tree region. 6-16
Figure 6-7 : Flat Spanning Tree Example.. 6-21
Figure 6-8 : Per VLAN (single and 802.1Q) Spanning Tree Example.. 6-22
Figure 6-9 : Automatic VLAN Containment - AVC not enabled.. 6-32
Figure 6-10 : Automatic VLAN Containment - AVC enabled.. 6-32
Figure 6-11 : Example Active Spanning Tree Topology. 6-43
Figure 6-12 : Sample MST Region Configuration. 6-46
Figure 6-13 : Flat Mode MSTP Quick Steps Example. 6-48
Figure 6-14 : Flat Mode MSTP with Superior MSTI 1 PPC Values. 6-50
Figure 7-1 : SPBM Network Components. 7-6
Figure 7-2 : Spanning Tree Topology.. 7-7
Figure 7-3 : ISIS-SPB Shortest Path Calculations. 7-9
Figure 7-4 : ISIS-SPB Topology.. 7-10
Figure 7-5 : Sample SPBM Network. 7-12

Figure 7-6 : Local and Remote SPB Pseudo-Wire Connections. 7-14
Figure 7-7 : RFP in a Sample SPBM Network. 7-16
Figure 7-8 : L3 VPN Interface: Inline Routing (Service-Based IP Interface). 7-20
Figure 7-9 : L3 VPN Interface: Inline Routing (Front-Panel Ports). 7-20
Figure 7-10 : L3 VPN Interface: External Loopback. 7-21
Figure 7-11 : SPB Backbone over a Service Provider Network. 7-23
Figure 7-12 : Network Interpretation of SPB Pseudo-Node. 7-25
Figure 7-13 : SPB In-Band Management - Route Redistribution. 7-26
Figure 7-14 : Remote SAP Pseudo-Wire Connection.. 7-65
Figure 7-15 : Devices Remotely Connected on a Pseudo-Wire. 7-65
Figure 7-16 : RFP for SPB Example.. 7-70
Figure 7-17 : IPv4 L3 VPN Service-based Interfaces (Inline Routing). 7-82
Figure 7-18 : L3 VPN Front-Panel Ports (Inline Routing). 7-86
Figure 7-19 : IPv4 L3 VPN External Loopback and Service-based Interfaces. 7-89
Figure 7-20 : IPv4 Inter-ISID Routing Example (One VRF). 7-91
Figure 7-21 : IPv6 Inter-ISID Routing Example (One VRF). 7-93
Figure 7-22 : IPv4 Inter-ISID Routing Example (Two VRFs). 7-95
Figure 7-23 : IPv6 Inter-ISID Routing Example (Two VRFs). 7-97
Figure 7-24 : L3 VPN Interface Before Migration to Inline Routing. 7-99
Figure 7-25 : L3 VPN Interface After Migration to Inline Routing. 7-100
Figure 7-26 : SPB Backbone over a Shared Network.. 7-105
Figure 7-27 : SPB Backbone over Another SPB Network. 7-106
Figure 8-1 : Remote-origin LBD Overview. 8-5
Figure 8-2 : LBD Packet Processing Mechanism for LBD Service Access Ports - Scenario 1. 8-10
Figure 8-3 : LBD Packet Processing Mechanism for LBD Service Access Ports - Scenario 2. 8-11
Figure 9-1 : Example of a Static Link Aggregate Group Network. 9-5
Figure 9-2 : Sample Network Using Static Link Aggregation. 9-10
Figure 10-1 : Example of a Dynamic Aggregate Group Network. 10-6
Figure 10-2 : Sample Network Using Dynamic Link Aggregation. 10-25
Figure 11-1 : DHL Active-Active Operation. 11-3
Figure 11-2 : Dual-Home Link Active-Active Example. 11-8
Figure 11-3 : Recommended DHL Active-Active Topology. 11-10
Figure 11-4 : Unsupported DHL Active-Active Topology (Network Loops). 11-11
Figure 12-1 : Normal Mode. 12-5
Figure 12-2 : Protection Mode. 12-6

Figure 12-3 : ERPv2 on Multi Ring and Ladder Network with RPLs and Shared Links. 12-7
Figure 12-4 : Example ERP Overview. 12-21
Figure 12-5 : Example ERPv2 Overview. 12-23
Figure 12-6 : Access Port and Access Port Example. 12-28
Figure 12-7 : Access Port and SDP/Tunnel Example. 12-29
Figure 13-1 : Basic MRP Ring Topology and MRP Ring States. 13-3
Figure 14-1 : Initial Configuration of MVRP. 14-4
Figure 14-2 : Dynamic Learning of VLANs 10, 20, and 30. 14-5
Figure 14-3 : Dynamic Learning of VLAN 50. 14-6
Figure 15-1 : Enabling and Disabling Proprietary TLV. 15-12
Figure 15-2 : Application Example - LLDP MED. 15-14
Figure 16-1 : Using SIP Snooping. 16-6
Figure 16-2 : SIP Snooping Use Case. 16-12
Figure 17-1 : IP Forwarding. 17-6
Figure 17-2 : Dynamic Proxy ARP. 17-16
Figure 17-3 : Denial of Service (DoS) Filtering-1. 17-28
Figure 17-4 : Denial of Service (DoS) Filtering-2. 17-29
Figure 17-5 : Denial of Service (DoS) Filtering-3. 17-29
Figure 17-6 : Tunneling Operation. 17-38
Figure 18-1 : Example Multiple VRF Configuration. 18-6
Figure 19-1 : Basic traffic flow between IPv6 hosts over IPv4 domain. 19-10
Figure 19-2 : Basic traffic flow between native IPv6 hosts and 6 to 4 sites. 19-11
Figure 20-1 : IP Packet in IPsec Transport Mode. 20-5
Figure 20-2 : IP Packet protected by ESP. 20-6
Figure 20-3 : IP Packet protected by AH. 20-7
Figure 20-4 : ESP Between Two OmniSwitches. 20-20
Figure 20-5 : Discarding RIPng Packets. 20-22
Figure 21-1 : RIP Routing. 21-6
Figure 22-1 : Example OSPF Network using the BFD Protocol. 22-33
Figure 23-1 : DHCP Clients are Members of the Same VLAN. 23-6
Figure 23-2 : DHCP Clients in Two VLANs. 23-7
Figure 23-3 : Sample SPB Network Topology with DHCP Relay. 23-11
Figure 24-1 : Internal DHCP Server Application Example. 24-12
Figure 25-1 : VRRP Redundancy Example. 25-5
Figure 25-2 : IPv4 VRRP Redundancy and Load Balancing. 25-26

Figure 25-3 : VRRP Tracking Example. 25-27
Figure 25-4 : IPv6 VRRP Redundancy and Load Balancing. 25-29
Figure 25-5 : VRRP Tracking Example. 25-30
Figure 26-1 : Example of a Server Load Balancing (SLB) Cluster. 26-7
Figure 26-2 : Weighted Round Robin Algorithm. 26-8
Figure 27-1 : Example of an IPMS Network. 27-4
Figure 27-2 : IPMSv6 Example. 27-28
Figure 27-3 : Example IPMS Network. 27-45
Figure 27-4 : Example IPMS v6 Network. 27-47
Figure 28-1 : Sample QoS Setup. 28-3
Figure 28-2 : Queue Set (QSet) Framework (Unicast Traffic). 28-12
Figure 28-3 : Basic Operation of Tri-Color Marking. 28-22
Figure 28-4 : Traffic Prioritization Example. 28-76
Figure 28-5 : Mapping Application. 28-80
Figure 28-6 : Routing All IP Source Traffic through a Firewall. 28-81
Figure 28-7 : Using a Built-In Port Group. 28-81
Figure 29-1 : Policy Server Setup. 29-3
Figure 30-1 : Access Guardian Overview. 30-12
Figure 30-2 : UNP - Device authentication and classification process. 30-26
Figure 30-3 : Campus Network Topology Example. 30-89
Figure 30-4 : Customized Captive Portal login page. 30-98
Figure 30-5 : Captive Portal - Login Screen. 30-99
Figure 30-6 : Captive Portal - Authentication is successful. 30-99
Figure 30-7 : OmniSwitch AP Discovery and Integration Example - VLAN Domain. 30-103
Figure 30-8 : OmniSwitch AP Discovery and Integration Example - SPB Domain. 30-104
Figure 30-9 : OmniSwitch L2 GRE Tunneling Configuration Example. 30-123
Figure 30-10 : Access Guardian Network Configuration Example. 30-130
Figure 30-11 : BYOD Network Illustration. 30-151
Figure 30-12 : Importing the Alcatel-Lucent Enterprise dictionary into CPPM. 30-156
Figure 30-13 : mDNS Work Flow. 30-162
Figure 30-14 : SSDP - Wired DLNA-Capable Client. 30-165
Figure 30-15 : SSDP - Wireless DLNA-Capable Clients. 30-166
Figure 30-16 : Zero Configuration Networking - Aruba Mode. 30-168
Figure 30-17 : Gateway Mode Setup. 30-169
Figure 30-18 : mDNS Service Sharing. 30-169

Figure 30-19 : mDNS Gateway - mDNS Devices Connected to an SPB Domain. 30-171
Figure 30-20 : Responder Mode Setup. 30-172
Figure 30-21 : mDNS Responder - mDNS Devices Connected to an SPB Domain. 30-173
Figure 30-22 : BYOD Network with Employee and Guest Devices. 30-181
Figure 30-23 : IoT Device Profiling Overview.. 30-198
Figure 30-24 : IoT Example - OV interaction with OmniSwitch. 30-202
Figure 31-1 : Application Monitoring Process. 31-4
Figure 31-2 : Application Enforcement Process. 31-5
Figure 32-1 : AFP Overview.. 32-5
Figure 33-1 : Servers Used for Authenticated Switch Access. 33-6
Figure 33-2 : Directory Information Tree. 33-26
Figure 33-3 : Kerberos Snooping Authentication. 33-41
Figure 34-1 : Example Port Mapping Topology.. 34-6
Figure 35-1 : How LPS Authorizes Source MAC Addresses.. 35-7
Figure 36-1 : Relationship Between Mirrored and Mirroring Ports. 36-11
Figure 36-2 : Port Mirroring Using External RMON Probe.. 36-12
Figure 36-3 : Remote Port Mirroring Example.. 36-18
Figure 36-4 : Policy Based Multiple Destination Mirroring.. 36-19
Figure 36-5 : RPMIR Loopback Session Configuration.. 36-21
Figure 36-6 : Port Mirroring Using External RMON Probe.. 36-33
Figure 36-7 : Monitoring Resource Availability from Multiple Ports and Switches. 36-40
Figure 37-1 : VLAN Stacking Elements.. 37-4
Figure 37-2 : VLAN Stacking Application.. 37-27
Figure 37-3 : Outward (Egress) Loopback Test Example. 37-33
Figure 37-4 : Inward (Ingress) Loopback Test. 37-34
Figure 39-1 : Ethernet OAM - CFM Maintenance Domain. 39-4
Figure 40-1 : Example LINK OAM. 40-4
Figure 41-1 : CPE Test Head Example - Unidirectional, Ingress Test. 41-5
Figure 41-2 : Configuring a CPE Test Profile. 41-7
Figure 41-3 : CPE Test group Example - Unidirectional, Ingress Test. 41-17
Figure 41-4 : Configuring a CPE Test Group Profile. 41-20
Figure 41-5 : Sample Unidirectional Test Configuration. 41-27
Figure 41-6 : Sample Bidirectional Test Configuration. 41-28
Figure 41-7 : Sample Bidirectional Multi-stream Test Configuration. 41-29
Figure 42-1 : Network overview for PPPoE IA. 42-4

About This Guide

This *OmniSwitch AOS Release 8 Network Configuration Guide* describes basic attributes of your switch and basic switch administration tasks. The software features described in this manual are shipped standard with your switches. These features are used when readying a switch for integration into a live network environment.

Supported Platforms

The information in this guide applies only to the following products:

- OmniSwitch 6360 Series
- OmniSwitch 6465 Series
- OmniSwitch 6560 Series
- OmniSwitch 6860 Series
- OmniSwitch 6865 Series
- OmniSwitch 6900 Series
- OmniSwitch 9900 Series

Who Should Read this Manual?

The audience for this user guide are network administrators and IT support personnel who need to configure, maintain, and monitor switches and routers in a live network. However, anyone wishing to gain knowledge on how fundamental software features are implemented in the OmniSwitch Series switches will benefit from the material in this configuration guide.

When Should I Read this Manual?

Read this guide as soon as your switch is up and running and you are ready to familiarize yourself with basic software functions. You should have already stepped through the first login procedures and read the brief software overviews in the appropriate Hardware Users Guide.

You should have already set up a switch password and be familiar with the very basics of the switch software. This manual will help you understand the switch's directory structure, the Command Line Interface (CLI), configuration files, basic security features, and basic administrative functions. The features and procedures in this guide will help form a foundation that will allow you to configure more advanced switching features later.

What is in this Manual?

This configuration guide includes information about the following features:

- Basic switch administrative features, such as file editing utilities, procedures for loading new software, and setting up system information (name of switch, date, time).
- Configurations files, including snapshots, off-line configuration, time-activated file download.
- The CLI, including on-line configuration, command-building help, syntax error checking, and line editing.
- Basic security features, such as switch access control and customized user accounts.
- SNMP
- Web-based management (WebView)

What is Not in this Manual?

The configuration procedures in this manual primarily use Command Line Interface (CLI) commands in examples. CLI commands are text-based commands used to manage the switch through serial (console port) connections or via Telnet sessions. This guide does include introductory chapters for alternative methods of managing the switch, such as web-based (WebView) and SNMP management. However the primary focus of this guide is managing the switch through the CLI.

Further information on WebView can be found in the context-sensitive on-line help available with that application.

This guide does not include documentation for the OmniVista network management system. However, OmniVista includes a complete context-sensitive on-line help system.

This guide provides overview material on software features, how-to procedures, and tutorials that will enable you to begin configuring your OmniSwitch. However, it is not intended as a comprehensive reference to all CLI commands available in the OmniSwitch. For such a reference to all CLI commands, consult the *OmniSwitch AOS Release 8 CLI Reference Guide*.

How is the Information Organized?

Each chapter in this guide includes sections that will satisfy the information requirements of casual readers, rushed readers, serious detail-oriented readers, advanced users, and beginning users.

Quick Information. Most chapters include a *specifications table* that lists RFCs and IEEE specifications supported by the software feature. In addition, this table includes other pertinent information such as minimum and maximum values and sub-feature support. Some chapters include a *defaults table* that lists the default values for important parameters along with the CLI command used to configure the parameter. Many chapters include *Quick Steps* sections, which are procedures covering the basic steps required to get a software feature up and running.

In-Depth Information. All chapters include *overview sections* on software features as well as on selected topics of that software feature. *Topical sections* may often lead into *procedure sections* that describe how to configure the feature just described. Many chapters include *tutorials* or *application examples* that help convey how CLI commands can be used together to set up a particular feature.

Documentation Roadmap

The OmniSwitch user documentation suite was designed to supply you with information at several critical junctures of the configuration process.The following section outlines a roadmap of the manuals that will help you at each stage of the configuration process. Under each stage, we point you to the manual or manuals that will be most helpful to you.

Stage 1: Using the Switch for the First Time

Pertinent Documentation: *OmniSwitch Hardware Users Guide*
Release Notes

This guide provides all the information you need to get your switch up and running the first time. It provides information on unpacking the switch, rack mounting the switch, installing NI modules, unlocking access control, setting the switch's IP address, and setting up a password. It also includes succinct overview information on fundamental aspects of the switch, such as hardware LEDs, the software directory structure, CLI conventions, and web-based management.

At this time you should also familiarize yourself with the Release Notes that accompanied your switch. This document includes important information on feature limitations that are not included in other user guides.

Stage 2: Gaining Familiarity with Basic Switch Functions

Pertinent Documentation: *OmniSwitch Hardware Users Guide*
OmniSwitch AOS Release 8 Switch Management Guide

Once you have your switch up and running, you will want to begin investigating basic aspects of its hardware and software. Information about switch hardware is provided in the *Hardware Guide*. This guide provide specifications, illustrations, and descriptions of all hardware components, such as chassis, power supplies, Chassis Management Modules (CMMs), Network Interface (NI) modules, and cooling fans. It also includes steps for common procedures, such as removing and installing switch components.

The *OmniSwitch AOS Release 8 Switch Management Guide* is the primary users guide for the basic software features on a single switch. This guide contains information on the switch directory structure, basic file and directory utilities, switch access security, SNMP, and web-based management. It is recommended that you read this guide before connecting your switch to the network.

Stage 3: Integrating the Switch Into a Network

Pertinent Documentation: *OmniSwitch AOS Release 8 Network Configuration Guide*
OmniSwitch AOS Release 8 Advanced Routing Configuration Guide
OmniSwitch AOS Release 8 Data Center Switching Guide

When you are ready to connect your switch to the network, you will need to learn how the OmniSwitch implements fundamental software features, such as 802.1Q, VLANs, Spanning Tree, and network routing protocols. The *OmniSwitch AOS Release 8 Network Configuration Guide* contains overview information, procedures, and examples on how standard networking technologies are configured on the OmniSwitch.

The *OmniSwitch AOS Release 8 Advanced Routing Configuration Guide* includes configuration information for networks using advanced routing technologies (OSPF and BGP) and multicast routing protocols (DVMRP and PIM-SM).

The *OmniSwitch AOS Release 8 Data Center Switching Guide* includes configuration information for data center networks using virtualization technologies, such as Data Center Bridging (DCB) protocols, Virtual eXtensible LAN (VxLAN), and Fibre Channel over Ethernet (FCoE) network convergence.

Anytime

The *OmniSwitch AOS Release 8 CLI Reference Guide* contains comprehensive information on all CLI commands supported by the switch. This guide includes syntax, default, usage, example, related CLI command, and CLI-to-MIB variable mapping information for all CLI commands supported by the switch. This guide can be consulted anytime during the configuration process to find detailed and specific information on each CLI command.

Related Documentation

The following are the titles and descriptions of all the related OmniSwitch user manuals:

- *OmniSwitch 6360, 6465, 6560, 6860, 6865, 6900, 9900 Hardware Users Guides*

  Describes the hardware and software procedures for getting an OmniSwitch up and running as well as complete technical specifications and procedures for all OmniSwitch chassis, power supplies, fans, and Network Interface (NI) modules.

- *OmniSwitch AOS Release 8 CLI Reference Guide*

  Complete reference to all CLI commands supported on the OmniSwitch. Includes syntax definitions, default values, examples, usage guidelines and CLI-to-MIB variable mappings.

- *OmniSwitch AOS Release 8 Switch Management Guide*

  Includes procedures for readying an individual switch for integration into a network. Topics include the software directory architecture, image rollback protections, authenticated switch access, managing switch files, system configuration, using SNMP, and using web management software (WebView).

- *OmniSwitch AOS Release 8 Network Configuration Guide*

  Includes network configuration procedures and descriptive information on all the major software features and protocols included in the base software package. Chapters cover Layer 2 information (Ethernet and VLAN configuration), Layer 3 information (routing protocols, such as RIP and IPX), security options (authenticated VLANs), Quality of Service (QoS), link aggregation, and server load balancing.

- *OmniSwitch AOS Release 8 Advanced Routing Configuration Guide*

  Includes network configuration procedures and descriptive information on all the software features and protocols included in the advanced routing software package. Chapters cover multicast routing (DVMRP and PIM-SM), Open Shortest Path First (OSPF), and Border Gateway Protocol (BGP).

- *OmniSwitch AOS Release 8 Data Center Switching Guide*

  Includes an introduction to the OmniSwitch data center switching architecture as well as network configuration procedures and descriptive information on all the software features and protocols that support this architecture. Chapters cover Data Center Bridging (DCB) protocols, Virtual Network Profile (vNP), VxLAN, and FCoE/FC transit and gateway functionality.

- *OmniSwitch AOS Release 8 Transceivers Guide*

  Includes SFP and XFP transceiver specifications and product compatibility information.

- *OmniSwitch AOS Release 8 Specifications Guide*

  Includes Specifications table information for the features documented in the Switch Management Guide, Network Configuration Guide, Advanced Routing Guide, and Data Center Switching Guide.

- Technical Tips, Field Notices

  Includes information published by Alcatel-Lucent Enterprise's Customer Support group.

- *Release Notes*

  Includes critical Open Problem Reports, feature exceptions, and other important information on the features supported in the current release and any limitations to their support.

Technical Support

An Alcatel-Lucent Enterprise service agreement brings your company the assurance of 7x24 no-excuses technical support. You'll also receive regular software updates to maintain and maximize your Alcatel-Lucent Enterprise product's features and functionality and on-site hardware replacement through our global network of highly qualified service delivery partners.

With 24-hour access to Alcatel-Lucent Enterprise's Service and Support web page, you'll be able to view and update any case (open or closed) that you have reported to Alcatel-Lucent Enterprise's technical support, open a new case or access helpful release notes, technical bulletins, and manuals.

Access additional information on Alcatel-Lucent Enterprise's Service Programs:

Web: https://myportal.al-enterprise.com

Phone: 1-800-995-2696

Email: ebg_global_supportcenter@al-enterprise.com

1 Configuring Ethernet Ports

The Ethernet software is responsible for a variety of functions that support Ethernet ports on OmniSwitch Series switches. These functions include diagnostics, software loading, initialization, configuration of line parameters, gathering statistics, and responding to administrative requests from SNMP or CLI.

In This Chapter

This chapter describes the Ethernet port parameters of the switch and how to configure them through the Command Line Interface (CLI). CLI Commands are used in the configuration examples.

Configuration procedures described in this chapter include:

- "Configuring Ethernet Port Parameters" on page 1-3
- "Using TDR Cable Diagnostics" on page 1-12
- "Interfaces Violation Recovery" on page 1-14
- "Clearing Ethernet Port Violations" on page 1-18
- "Link Monitoring" on page 1-19
- "Link Fault Propagation" on page 1-23
- "IEEE 1588 Precision Time Protocol (PTP)" on page 1-26
- "MAC Security Overview" on page 1-28

For more information about using CLI commands to view Ethernet port parameters, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Ethernet Port Defaults

The following table shows Ethernet port default values:

Parameter Description	Command	Default Value/Comments
Interface Line Speed	**interfaces speed**	AutoNeg
Interface Duplex Mode	**interfaces duplex**	AutoNeg
Trap Port Link Messages	**interfaces link-trap**	Disabled
Interface Configuration	**interfaces**	Enabled
Peak Flood Rate Configuration	**interfaces flood-limit**	4 Mbps (10M Ethernet) 49 Mbps (100M Ethernet) 496 Mbps (1G Ethernet) 700 Mbps (2.5G Ethernet) 997 Mbps (10G Ethernet) 997 Mbps (40G Ethernet) 997 Mbps (100G Ethernet)
Interface Alias	**interfaces alias**	None configured
Maximum Frame Size	**interfaces max-frame-size**	1553 (untagged) Ethernet packets 1553 (tagged) Ethernet packets 9216 Gigabit Ethernet packets
Digital Diagnostics Monitoring (DDM)	**interfaces ddm**	Disabled
Enhanced Port Performance (EPP)	**interfaces**	Disabled
Beacon LED	**interfaces beacon**	Disabled
Precision Time Protocol (PTP) time stamping on the switch	**interfaces ptp admin-state**	Disabled

Ethernet Ports Overview

This chapter describes the Ethernet software CLI commands used for configuring and monitoring the Ethernet port parameters of your switch.

Configuring Ethernet Port Parameters

The following sections describe how to use CLI commands to configure ethernet ports.

Enabling and Disabling Autonegotiation

To enable or disable autonegotiation on a single port, a range of ports, or an entire slot, use the **interfaces** command. For example:

```
-> interfaces 2/3 autoneg enable
-> interfaces 2/1-3 autoneg enable
-> interfaces 2 autoneg enable
```

Configuring Crossover Settings

To configure crossover settings on a single port, a range of ports, or an entire slot, use the **interfaces crossover** command. If autonegotiation is disabled, auto MDIX, auto speed, and auto duplex are not accepted.

Setting the crossover configuration to **auto** configures the interface or interfaces to automatically detect crossover settings. Setting crossover configuration to **mdix** configures the interface or interfaces for MDIX (Media Dependent Interface with Crossover), which is the standard for hubs and switches. Setting crossover to **mdi** configures the interface or interfaces for MDI (Media Dependent Interface), which is the standard for end stations.

For example:

```
-> interfaces 2/1 crossover auto
-> interfaces 2/2-5 crossover mdi
-> interfaces 3 crossover mdix
```

Setting Interface Line Speed

The **interfaces speed** command is used to set the line speed on a specific port, a range of ports, or all ports on an entire slot.

For example:

```
-> interfaces 2/1 speed 100
-> interfaces 2/2-5 speed 1000
-> interfaces 3 speed auto
```

Configuring Duplex Mode

The **interfaces duplex** command is used to configure the duplex mode on a specific port, a range of ports, or all ports on a slot to **full, half, or auto**. (The **auto** option causes the switch to advertise all available duplex modes (half/full/both) for the port during autonegotiation.) In full duplex mode, the interface transmits and receives data simultaneously. In half duplex mode, the interface can only transmit or receive data at a given time.

For example:

```
-> interfaces 2/1 duplex half
-> interfaces 2/2-5 duplex auto
-> interfaces 3 duplex full
```

Setting Trap Port Link Messages

The **interfaces link-trap** command can be used to enable or disable trap port link messages on a specific port, a range of ports, or all ports on a slot. When enabled, a trap message is sent to a Network Management Station (NMS) whenever the port state has changed.

For example:

```
-> interfaces 2/3 link-trap enable
-> interfaces 2/3-5 link-trap enable
-> interfaces 2 link-trap enable
```

Resetting Statistics Counters

The **clear interfaces** command is used to reset all Layer 2 statistics counters on a specific port, a range of ports, or all ports on a slot. For example:

```
-> clear interfaces 2/3 l2-statistics
-> clear interfaces 2/1-3 l2-statistics
-> clear interfaces 2 l2-statistics
```

This command also includes an optional **cli** parameter. When this parameter is specified, only those statistics that are maintained by the switch CLI are cleared; SNMP values are not cleared and continue to maintain cumulative totals. For example:

```
-> clear interfaces 2/1-3 l2-statistics cli
```

Note that when the **cli** parameter is not specified both CLI and SNMP statistics are cleared.

Enabling and Disabling Interfaces

The **interfaces** command is used to enable or disable a specific port, a range of ports, or all ports on an entire slot.

```
-> interfaces 2/3 admin-state disable
-> interfaces 2/1-3 admin-state disable
-> interfaces 2 admin-state disable
```

Configuring a Port Alias

The **interfaces alias** command is used to configure an alias (i.e., description) for a single port. (You cannot configure an entire switch or a range of ports.) For example:

```
-> interfaces 2/3 alias ip_phone1
-> interfaces 2/3 alias "ip phones 1"
```

Note. Spaces must be contained within quotes.

Configuring Maximum Frame Sizes

The **interfaces max-frame-size** command can be used to configure the maximum frame size (in bytes) on a specific port, a range of ports, or all ports on a switch.

For example:

```
-> interfaces 2/3 max frame 9216
-> interfaces 2/1-3 max frame 9216
-> interfaces 2 max frame 9216
```

Configuring Digital Diagnostic Monitoring (DDM)

Digital Diagnostics Monitoring allows the switch to monitor the status of a transceiver by reading the information contained on the transceiver's EEPROM. The transceiver can display Actual, Warning-Low, Warning-High, Alarm-Low and Alarm-High for the following:

- Temperature
- Supply Voltage
- Current
- Output Power
- Input Power

To enable the DDM capability on the switch use the **interfaces ddm** command. For example, enter:

```
-> interfaces ddm enable
```

Traps can be enabled using the **interfaces ddm-trap** if any of the above values crosses the pre-defined low or high thresholds of the transceiver. For example:

```
-> interfaces ddm-trap enable
```

Note. In order to take advantage of the DDM capability, the transceiver must support the DDM functionality. Not all transceivers support DDM; refer to the Transceivers Guide for additional DDM information.

Configuring Flood Rate Limiting

The OmniSwitch implementation of storm control supports flood rate limiting for broadcast, unknown unicast, and multicast traffic. A high threshold rate is configured in megabits-per-second (mbps), packets-per-second (pps), or as a percentage of the port speed. When the threshold value is reached, packets are dropped.

To configure the flood rate limit threshold, use the **interfaces flood-limit** command For example:

```
-> interfaces 2/1/1 flood-limit bcast rate mbps 100
-> interfaces 2/1/2-5 flood-limit uucast rate pps 500
-> interfaces slot 3/1 flood-limit mcast rate cap% 50
```

Configuring a Flood Rate Limit Action

Configuring a port shutdown or trap action to occur when the rate limit threshold is reached provides a method for monitoring storm traffic.

- Port shutdown action—port is moved to a STORM violated state and a violation trap is sent.
- Trap action—the storm is controlled through flood rate limiting and a trap is sent. The port is not moved into a STORM violated state.

To configure the flood rate limit action, use the **interfaces flood-limit action** command with either the **shutdown** or **trap** option. For example:

```
-> interfaces 1/1/1 flood-limit bcast action shutdown
-> interfaces 1/1/4 flood-limit uucast action trap
```

Use the **all** option with the **interfaces flood-limit action** command to specify all types of traffic. For example:

```
-> interfaces 1/1/11 flood-limit all action shutdown
```

To set the flood rate limit action back to the default value (no action is taken, packets above the threshold are dropped) use the **interfaces flood-limit action** command with the **default** option. For example:

```
-> interfaces 1/1/14 flood-limit mcast action default
```

Configuring Auto-Recovery for Port Shutdown Action

When a port is shutdown because of a STORM violated state, an administrator will have to clear the violation. However, configuring a low threshold value for flood rate limiting can help to automate this process. When the rate of violating traffic received on the port goes below the low threshold value, the port is removed from the violating state.

To configure the low threshold value, use the **interfaces flood-limit** command with the **low-threshold** option. For example:

```
-> interfaces 1/1/1 flood-limit bcast rate mbps 60 low-threshold 40
-> interfaces 1/1/4 flood-limit uucast rate mbps 100 low-threshold 40
-> interfaces 1/1/5 flood-limit mcast rate pps 2000 low-threshold 1000
```

Configuring Flow Control

The **interfaces pause** command is used to configure flow control (pause) settings for ports that run in full duplex mode. Configuring flow control is done to specify whether or not an interface transmits, honors, or both transmits and honors PAUSE frames. PAUSE frames are used to temporarily pause the flow of traffic between two connected devices to help prevent packet loss when traffic congestion occurs between switches.

Note that if autonegotiation and flow control are both enabled for an interface, then autonegotiation determines how the interface processes PAUSE frames. If autonegotiation is disabled but flow control is enabled, then the configured flow control settings apply.

By default, flow control is disabled. To configure flow control for one or more ports, use the **interfaces pause** command with one of the following parameters to specify how PAUSE frames are processed:

- **tx—**Transmit PAUSE frames to peer switches when traffic congestion occurs on the local interface. Do not honor PAUSE frames from peer switches.
- **rx—**Allow the interface to honor PAUSE frames from peer switches and temporarily stop sending traffic to the peer. Do not transmit PAUSE frames to peer switches.
- **tx-and-rx—**Transmit and honor PAUSE frames when traffic congestion occurs between peer switches.

For example, the following command configures ports 1/1 through 1/10 to transmit and honor PAUSE frames:

```
-> interfaces 1/1-10 pause tx-and-rx
```

To disable flow control for one or more ports, specify the **disable** parameter with the **interfaces pause** command. For example:

```
-> interfaces 1/10 pause disable
```

Enabling and Disabling Enhanced Port Performance (EPP)

EPP can assist in connecting with SFF-8431 non-compliant or electrically deficient devices. EPP can be used on some links to enhance the receive signal sampling resolution management and help to improve the link integrity to the link partner. The following steps should be followed to determine if EPP should be enabled:

1 **Check the current link quality** - Check the current link quality of the interface.

2 **Diagnose any link quality issues** - If the Link Quality is not '**Good**'. Perform a few basic troubleshooting steps to determine if the issue is with the link partner and whether enabling EPP can help improve the quality.

3 **Enable EPP** - If it's determined that the issue is with the link parter, enable EPP.

EPP - Product and Transceiver Support

Only certain transceivers support enabling EPP. Additionally, depending on the revision of the OmniSwitch, there are port restrictions due to the power requirements of enabling EPP as shown in the table below.

Product	Rev	EPP Support
OS6900-X20	B11 B10 or less	No restriction Only 5 ports can have EPP enabled
OS6900-X40	B11 B10 or less	No restriction Only 5 ports in 1st group of 20 and 5 ports in 2nd group of 20
Expansion Board	Any	No restrictions
10-Gigabit Transceivers	N/A	Supported
1/40-Gigabit Transceivers	N/A	Not Supported

Product/Transceiver Support

EPP - Check the Current Link Quality

A Link-Quality parameter has been added to help support EPP functionality. If connectivity issues are being observed check the current link quality using the **interfaces** command and observe the EPP output. For example:

```
-> show interfaces 2/1
(output truncated)

EPP                     : Disabled,   Link-Quality:Fair
```

Link-Quality	Description
Good	Link is good
Fair	Link may exhibit errors
Poor	Link will exhibit errors and may lose connectivity
N/A	Link does not support EPP

EPP - Diagnose

For ports diagnosed as **Fair** or **Poor**, simple steps can be performed to identify the faulty component. Since the issue could be with the transceiver, cable, fiber, or the link partner, see the table below to help determine if the issue is with the link partner and if enabling EPP may help.

Media Type	Diagnostic Action
Direct Attached Copper Cable	• Disconnect cable from link partner • Connect free cable end to unused port of OS6900 • View the Link-Quality **Good** - The link partner should be diagnosed and enabling EPP may help. **Fair** or **Poor** - The direct-attached copper cable should be replaced.
SFP+ optical transceiver	• Replace SFP+ transceiver on OS6900 port • View the Link-Quality **Good** - The original SFP+ transceiver is faulty. **Fair** or **Poor** - The fiber cable or link partner should be diagnosed and enabling EPP may help.

EPP - Enabling

If after diagnosing the problem it is determined that the issue is with the link partner and the Link-Quality has been diagnosed to be **Fair** or **Poor,** EPP can be enabled allowing the system to operate with the deficient receive channels. For example:

```
-> interfaces 2/1 epp enable
```

After enabling EPP continue to monitor the Link-Quality.

Configuring Energy Efficient Ethernet (802.3az)

Energy Efficient Ethernet (EEE) is a protocol to allow ports to operate in idle or low power mode when there is no traffic to send. When EEE is enabled on a port it will advertise its EEE capability to its link partner. If the partner supports EEE they will operate in EEE mode. If the partner does not support EEE the ports will operate in legacy mode. This allows EEE capable switches to be deployed in existing networks avoiding backward compatibility issues.

- EEE is only applicable to 10GBase-T ports.
- The LLDP option in IEEE 802.3az standard is not currently supported.

To enable the EEE capability on the switch use the **interfaces eee** command. For example, enter:

```
-> interfaces 1/1 eee enable
```

Configuring Split-Mode

Some OmniSwitch models support 4X10G splitter cables to allow a 40G port to be configured as four 10G ports. The **interfaces primary-port split-mode** command is used to configure the mode (**auto, 40G, 4X10G**).

When a splitter cable is used the port numbering scheme changes to accommodate the four 10G ports by using letters a, b, c, d to refer to the 10G sub-ports. When referring to a single sub-port the port letter should be used to differentiate between all the sub-ports. If no letter is given the command assumes port 'a', for example.

```
-> show interfaces 1/1/1 - refers to interface 1/1/1a
-> show interfaces 1/1/1a - refers to interface 1/1/1a
-> show interfaces 1/1/1d - refers to interface 1/1/1d
```

When referring to a range of ports the lettered sub-ports are implied, for example:

```
-> show interfaces 1/1/1-2 - refers to interfaces 1/1/1a, 1b, 1c, 1d and 1/1/2a,
2b, 2c, 2d
-> show interfaces 1/1/1a-1c - refers to interfaces 1/1/1a, 1b, 1c
-> show interfaces 1/1/1-2a - refers to interfaces 1/1/1a, 1b, 1c, 1d, and 1/1/
2a.
```

Configuring Beacon LED

The Beacon LED feature provides a mechanism to allow an administrator to configure the color and the mode of a port LED using the **interfaces beacon** command. This can useful in the following scenarios:

- Port identification: Can help to identify a particular port(s) needing attention or where a cable may need to be swapped. Manually changing the color or mode of the port LED can help to guide a technician to a particular port. This can also be helpful in a highly dense mesh of cabling.
- Power Savings: Large Data Centers are looking for ways to reduce power consumption. One way could be to power off every LED on every node if operating properly and only use the LEDs for indicating ports that need attention.
- Tracking link activity: Servers are often configured in clusters for certain functions or applications. Ports could be color coded to differentiate between clusters.

Note. The beacon LED feature does not affect the normal behavior of switch ports or traffic flow. It only sets LED colors and behaviors for the uses listed above. If an actual alarm or issue is detected on the switch, important LED status information related to the issue takes precedence and overrides beacon settings.

Note. The beacon LED feature is not supported on sub-ports 'b', 'c', or 'd' when an interface is operating in 4X10G mode. Additionally, only Solid mode is supported on sub-port 'a' for 4X10G interfaces.

LED Color and Mode Settings

- LED Color - The color of the LED can be changed to yellow, white, red, magenta, green, blue, aqua, or off.
- Activity Mode - The LED will blink normally based on the port activity but the color of the LED can be changed.
- Solid Mode - The LED will not blink based on the port activity, it will always be solid. The color of the LED can be changed.

```
-> interfaces 1/1/30 beacon admin-status enable

-> interfaces 1/1/30 beacon led-color magenta

-> interfaces 1/1/30 beacon led-mode solid
```

Beacon settings are saved and can be administratively enabled or disabled as needed at a later time. This allows administrators to reuse previously configured LED settings without having to start over. Use the **show interfaces beacon** command to view the beacon settings.

```
-> show interfaces beacon
 Ch/Slot/Port   Admin-Stat   LED-Color    LED-Mode
------------+-------------+--------------+------------
 1/1/1A          Disable     Magenta      Solid
 1/1/2A          Disable     Blue         Activity
 1/1/30A         Enable      Magenta      Solid
 1/1/31A         Enable      Off          Solid
```

Using TDR Cable Diagnostics

Time Domain Reflectometry (TDR) is a feature that is used to detect cable faults. This feature is best deployed in networks where service providers and system administrators want to quickly diagnose the state of a cable during outages, before proceeding with further diagnosis.

When a TDR test is initiated, a signal is sent down a cable to determine the distance to a break or other discontinuity in the cable path. The length of time it takes for the signal to reach the break and return is used to estimate the distance to the discontinuity.

Initiating a TDR Cable Diagnostics Test

Consider the following guidelines before initiating a TDR test:

- Only one test can run at any given time, and there is no way to stop a test once it has started.
- The TDR test runs an “out-of-service” test; other data and protocol traffic on the port is interrupted when the test is active.
- TDR is supported only on copper ports.
- TDR is not supported on Link aggregate ports.
- Each time a TDR test is run, statistics from a test previously run on the same port are cleared.

A TDR test is initiated using the **interfaces tdr** CLI command. For example, the following command starts the test on port 2/1:

```
-> interfaces 1/1/1 tdr enable
```

Displaying TDR Test Results

The **show interfaces tdr-statistics** command is used to display TDR test statistics. For example:

```
-> show interfaces 1/1/1 tdr-statistics
Ch/
Slot/ No of Cable Fuzzy  Pair1 Pair1  Pair2 Pair2  Pair3 Pair3  Pair4 Pair4  Test
Port pairs  State Length State Length State Length State Length State Length Result
-----+-----+------+-----+-----+-----+------+------+-----+------+-----+------+------
1/1/1     4     ok     0    ok    3     ok      3     ok     3     ok    3     success
```

The following cable states are indicated in the **show interfaces tdr-statistics** command output:

- **OK**—Wire is working properly
- **Open**:—Wire is broken
- **Short**—Pairs of wire are in contact with each other
- **Crosstalk**—Signal transmitted on one pair of wire creates an undesired effect in another wire.
- **Unknown**:—Cable diagnostic test unable to find the state of a cable.

Clearing TDR Test Statistics

The **clear interfaces** command is used to clear the statistics of the last test performed on the port. There is no global statistics clear command. For example, the following command clears the TDR statistics:

```
-> clear interfaces 1/1/1 tdr-statistics
```

TDR statistics from a previous test are also cleared when a new test starts on the same port.

Interfaces Violation Recovery

The OmniSwitch allows features to shutdown an interface when a violation occurs on that interface. To support this functionality, the following interfaces violation recovery mechanisms are provided:

- Manual recovery of a downed interface using the **interfaces primary-port split-mode** command.
- An automatic recovery timer that indicates how much time a port remains shut down before the switch automatically brings the port back up (see "Configuring the Violation Recovery Time" on page 1-16).
- A maximum number of recovery attempts setting that specifies how many recoveries can occur before a port is permanently shutdown (see "Configuring the Violation Recovery Time" on page 1-16).
- A wait-to-restore timer that indicates the amount of time the switch waits to notify features that the port is back up (see "Configuring the Wait-to-Restore Timer" on page 1-20).
- An SNMP trap that is generated each time an interface is shutdown by a feature. This can occur even when the interface is already shutdown by another feature. The trap also indicates the reason for the violation.
- An SNMP trap that is generated when a port is recovered. The trap also includes information about how the port was recovered. Enabling or disabling this type of trap is allowed using the **violation recovery-trap** command.

Violation Shutdown and Recovery Methods

A port can be shutdown with one of the following methods, depending on the feature.

Filtering – The port is blocked by applying filtering to discard all packets sent or received on the port. With this method the link LED of the port remains ON. A port in this state can be recovered using the following methods:

- Using the **interfaces primary-port split-mode** command to manually clear the violation.
- Automatic recovery when the interface recovery timer expires.
- Using the **interfaces alias** command to administratively disable and enable the interface.
- Disconnecting and reconnecting the interface link.
- A link down and link up event.

Administratively – A port is administratively disabled. With this method the LED does not remain ON. A port in this state can be recovered using only the following methods:

- Using the **interfaces primary-port split-mode** command to manually clear the violation.
- Automatic recovery when the interface recovery timer expires.
- Using the **interfaces alias** command to administratively disable and enable the interface.

Disconnecting/reconnecting the interface link or a link down/up event *will not* recover a port that was administratively disabled.

Interface Violation Exceptions

An interface violation is not applied to an interface when any of the following scenarios occur:

- An interface is already in a permanent shutdown state. In this case, the only method for recovery is to use the **interfaces primary-port split-mode** command.
- An interface is already shutdown by another feature.
- An interface is not operationally up.

Interaction With Other Features

The table below lists the features that use the interfaces violation recovery mechanisms, along with the violation reason and shutdown type.

Feature	Reason Code	Shutdown Type
BPDU Shutdown	STP	Discard
User Port Shutdown	QOS	Discard
Policy rule - port disable	QOS	Discard
LPS	LPS-D	Discard
LPS	LPS-S	Admin-Down
UDLD	UDLD	Admin-Down
NetSec	NetSec	Admin-Down
NI	NISup	Admin-Down
LLDP Rouge Detection	LLDP	Discard
Link Monitoring	LinkMon	Admin-Down
Link Fault Propagation	LFP	Admin-Down
Remote Fault Propagation	RFP	Admin-Down

Configuring Interface Violation Recovery

The following sections provide information about how to configure parameter values that apply to the interfaces violation recovery mechanisms.

Configuring the Violation Recovery Time

The violation recovery time specifies the amount of time the switch waits before automatically recovering a port that was shut down due to a violation. When the recovery timer expires, the interface is operationally re-enabled and the violation on the interface is cleared.

Consider the following when configuring the violation recovery time:

- The timer value does not apply to interfaces that are in a permanent shutdown state. A port in this state is only recoverable using the **clear violation** command.
- The interface violation recovery mechanism is not supported on link aggregates, but is supported on the link aggregate member ports.

By default, the automatic recovery time is set to 300 seconds. Use the **violation recovery-time** command to change the automatic recovery time value, which is configurable on a per-port or global basis. For example, the following commands set the violation recovery time to 600 seconds at the global level and to 200 seconds for port 2/1 on chassis 1:

```
-> violation recovery-time 600
-> violation port 1/2/1 recovery-time 200
```

The violation recovery time value configured for a specific interface overrides the global value configured for all switch interfaces. To set the port-level value back to the global value, use the **default** parameter with the **violation recovery-time** command. For example, the following command sets the violation recovery time for port 2/1 on chassis 1 back to the global value of 600:

```
-> violation port 1/2/1 recovery-time default
```

Configuring the Violation Recovery Maximum Attempts

The violation recovery maximum setting specifies the maximum number of recovery attempts allowed before a port is permanently shut down. This value increments by one whenever an interface recovers from a violation using the automatic recovery timer mechanism. When the number of recovery attempts exceeds this configured threshold, the interface is permanently shut down. The only way to recover a permanently shut down interface is to use the **clear violation** command.

The recovery mechanism tracks the number of recoveries within a fixed time window (FTW). The FTW = 2 * maximum recovery number * recovery timer. For example, if the maximum number of recovery attempts is set to 4 and the recovery timer is set to 5, the FTW is 40 seconds (2 * 4 * 5=40).

The **violation recovery-maximum** command is used to configure the maximum number of recovery attempts. This value is configurable on a per-port or global basis. For example, the following commands set the number of attempts to 3 at the global level and to 5 for port 2/1on chassis 1:

```
-> violation recovery-maximum 3
-> violation port 1/2/1 recovery-maximum 5
```

The maximum recovery attempts value configured for a specific interface overrides the global value configured for all switch interfaces. To set the port-level value back to the global value, use the **default** parameter with the **violation recovery-maximum** command. For example, the following command sets the number of recovery attempts for port 2/1 on chassis 1 back to the global value of 3:

```
-> violation port 1/2/1 recovery-maximum default
```

To disable the violation recovery maximum attempts mechanism, set the number of attempts to zero. For example:

```
-> violation recovery-maximum 0
-> violation port 1/2/1 recovery-maximum 0
```

Verifying the Interfaces Violation Recovery Configuration

Use the following **show** commands to verify the violation recovery configuration:

show interfaces	Displays the administrative status, link status, violations, recovery time, maximum recovery attempts and the value of the wait-to-restore timer.
show violation	Displays the address violations that occur on ports with LPS restrictions. This command displays a port violation for sticky port security when the maximum number of MAC address of the connected workstation that the switch learns.
show interfaces portgroup	Displays the globally configured recovery time, SNMP recovery trap enable/disable status and maximum recovery attempts.

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Clearing Ethernet Port Violations

The following switch applications may trigger a violation condition on one or more ports:

- Learned Port Security (LPS)
- Quality of Service (QoS)
- Network Security
- UniDirectional Link Detection (UDLD)
- Fabric stability related violations

Depending on the application and type of violation, specific actions are taken when a violation is detected on the port. For example, an application may take one of the following actions when the violation triggers a port shut down:

- **Admin Down**—deactivates the physical port.
- **Simulated Down**—the physical port shows as active but the applications are not allowed to access the port link. The port is put in a blocking state.

A security violation may occur under the following conditions:

- A port is configured as a secure port and the number of secure MAC addresses learned on the port has exceeded the maximum value.
- A device with a secure MAC address that is configured or learned on one of the secure ports attempts to access another secure port.

Consider the following regarding link aggregate security violations:

- When a violation occurs on a physical port that is a member of a link aggregate, the violation affects the entire link aggregate group. All ports on that link aggregate are either restricted or shut down.
- When the violations are cleared for the entire link aggregate group, the whole link aggregate group is reactivated.
- When a simulated down violation occurs, toggling the link clears the violation for both the link aggregates and physical ports.

To view the violation conditions that exist on individual ports or link aggregates, use the **show violation** command. For example:

```
-> show violation

 Port   Source     Action              Reason           Timer
------+----------+-------------------+----------------+--------
  1/1  src lrn    simulated down      lps shutdown     0
  1/2  src lrn    simulated down      lps restrict     0
    2  qos        admin down          policy           0
```

To clear all the MAC address violation logs and activate the port or link aggregate, use the **clear violation** command. For example:

```
-> clear violation port 1/10
-> clear violation linkagg 10-20
```

Link Monitoring

The Link Monitoring feature is used to monitor interface status to minimize the network protocol re-convergence that can occur when an interface becomes unstable. To track the stability of an interface, this feature monitors link errors and link flaps during a configured timeframe. If the number of errors or link flaps exceeds configured thresholds during this time frame, the interface is shut down.

Note. Link Monitoring can be enabled on the member ports of the link aggregate, but not on the entire link aggregate. Link Monitoring is not supported on the VFL ports.

There are no explicit Link Monitoring commands to recover a port from a Link Monitoring shutdown. See "Clearing Ethernet Port Violations" on page 1-18 for information of clearing port violations.

Monitoring Interface Errors

When physical errors occur on an interface, control and data traffic is dropped causing unnecessary re-convergence for the network protocol running on the interface. The Link Monitoring feature monitors the physical errors such as CRC, lost frames, error frames and alignment errors. When a configurable number of errors is detected within the duration of a link monitoring window, the interface is shut down.

To configure the number of errors allowed before the port is shut down, use the **interfaces link-monitoring link-error-threshold** command. For example:

```
-> interfaces 1/1 link-monitoring link-error-threshold 50
```

In this example, the port is shutdown if the number of link errors exceeds the threshold value of 50 during the link monitoring window timeframe.

Monitoring Interface Flapping

When physical connectivity errors occur on an interface, the interface becomes unstable and causes unnecessary re-convergence for the network protocols running on the interface. The Link Monitoring feature monitors these interface flaps and shuts down the interface when excessive flapping is detected.

- The shutdown action is a physical port shutdown (the PHY and LED are down).
- Whenever an interface comes up and it is not an administrative action (admin-up), the link flap counter is incremented.

The **interfaces link-monitoring link-flap-threshold** command is used to configure the number of flaps allowed before the interface is shutdown. For example:

```
-> interfaces 1/1 link-monitoring link-flap-threshold 5
```

In this example, the port is shutdown if the number of link flaps exceeds the threshold value of five during the link monitoring window timeframe.

Monitoring Window

The Link Monitoring window is a per-port configurable timer that is started whenever link-monitoring is enabled on a port. During this time frame interface receive errors and interface flaps are counted. If either of the values exceeds the configured thresholds the interface is shut down.

- The timer value can be modified even when the Link Monitoring timer is running and the new value of timer will take effect after the current running timer expires.
- The threshold values for link errors and link flaps can also be modified when link-monitoring timer is running; if the new threshold value is less than the current link-flap or link-error counter value, then the interface will be shutdown immediately.

The **interfaces link-monitoring time-window** command is used to configure the monitoring window timer. For example:

```
-> interfaces 1/1 link-monitoring time-window 500
```

In this example, link monitoring will monitor port 1/1 for 500 seconds.

Starting a Link Monitoring Session

The Link Monitoring window timer is started when the feature is enabled on an interface using the **interfaces link-monitoring admin-status** command. For example:

```
-> interfaces 1/1 link-monitoring admin-status enable
```

All the statistics (link errors and link flaps) for a port are reset to zero when Link Monitoring is enabled on that port.

Stopping a Link Monitoring Session

The Link Monitoring window timer is stopped when one of the following occurs:

- The **interfaces link-monitoring admin-status** command is used to disable the feature on the port. For example:

  ```
  -> interfaces 1/1 link-monitoring admin-status enable
  ```

- The port is shutdown by any feature, such as Link Monitoring, UDLD, or Link Fault Propagation.

Configuring the Wait-to-Restore Timer

The wait-to-restore (WTR) timer is used to implement a delay before an interface is made operational for other features. Only after the timer has expired will the interface become active allowing network protocols to converge more gracefully. The timer value is configured on a per-port basis and is started whenever one of the following link-up events occurs:

- An interface is administratively downed followed by administratively up.
- The **interfaces primary-port split-mode** command is used.
- An interface recovers from a violation due to the automatic recovery timer mechanism.
- An interface is made operationally up when the cable is plugged in.

Consider the following when configuring the wait-to-restore timer:

- If the interface goes down again while the WTR timer is still running, the WTR timer is stopped. Otherwise, the interface is recovered after the time expires.
- The WTR timer functionality has no impact on link-error or link-flap detection; these features are configurable even when the WTR timer is disabled.
- The timer value can be modified when the WTR timer is running; however, the new timer value does not take effect until after the current running timer expires.
- The WTR timer is reset on every link up event that is detected.
- The WTR timer is stopped on detection of every link down event.
- When the WTR timer is running, the interface is physically up but the link status is down.

The **interfaces wait-to-restore** command is used to configure the WTR timer value, in multiples of 5. For example, the following commands set the WTR timer value to 300 seconds:

```
-> interfaces 1/1 wait-to-restore 300
```

To disable the WTR timer mechanism, set the timer value to zero. For example:

```
-> interfaces 1/1 wait-to-restore 0
```

By default, the WTR time is disabled.

Configuring the Wait-to-Shutdown Timer

The wait-to-shutdown (WTS) timer is used to implement a delay before an interface is made non-operational for other applications such as data, control and management. Only after the timer has expired will the interface become disabled allowing network protocols to converge more gracefully. The timer value is configured on a per-port basis and is started whenever one of the following link-up events occurs:

- An interface is administratively brought down.
- An interface is shutdown from a violation.
- An interface is made operationally down when the cable is unplugged in.

When the interface goes down, the WTS timer will be started. If the interface comes back up while the WTS timer is running, then WTS timer will be stopped and no link down event will be sent. Otherwise, after the WTS timer expiries a link-down event will be sent to all the relevant applications.

Consider the following when configuring the wait-to-shutdown timer:

- If the interface comes back up while the WTS timer is still running, the WTS timer is stopped and no link down event is sent to other switch applications.
- The WTR timer functionality has no impact on link-error or link-flap detection; these features are configurable even when the WTS timer is disabled.
- The timer value can be modified when the WTS timer is running; however, the new timer value does not take effect until after the current running timer expires.
- When the wait-to-shutdown timer is running there would be packet loss on that interface. This is because the port is physical down and only the link-down event is not being communicated to the switch applications which will continue to send packets to the interface.

- The link-status of the remote connected port will be down when the WTS timer is running since the port is physically down.

The **interfaces wait-to-shutdown** command is used to configure the WTS timer value, in multiples of 10 milliseconds. For example, the following commands set the WTR timer value to 30 seconds:

```
-> interfaces 1/1 wait-to-shutdown 30000
```

To disable the WTR timer mechanism, set the timer value to zero. For example:

```
-> interfaces 1/1 wait-to-shutdown 0
```

By default, the WTS time is disabled.

Displaying Link Monitoring Information

Use the following **show** commands to display Link Monitoring statistics and configuration information:

show interfaces link-monitoring statistics	Displays Link Monitoring statistics, such as the link flap and error counts and the port state (shutdown, down, up).
show interfaces link-monitoring config	Displays the Link Monitoring configuration, such as the monitoring status, monitoring window time, and the link flap and error thresholds.
show interfaces	Displays the administrative status, link status, violations, recovery time, maximum recovery attempts and the value of the wait-to-restore timer.

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Link Fault Propagation

The Link Fault Propagation (LFP) feature provides a mechanism to propagate a local interface failure into another local interface. In many scenarios, a set of ports provide connectivity to the network. If all these ports go down, the connectivity to the network is lost. However, the remote end remains unaware of this loss of connectivity and continues to send traffic that is unable to reach the network. To solve this problem, LFP does the following:

- Monitors a group of interfaces (configured as source ports).
- If all the source ports in the group go down, LFP waits a configured amount of time then shuts down another set of interfaces (configured as destination ports) that are associated with the same group.
- When any one of the source ports comes back up, all of the destination ports are brought back up and network connectivity is restored.

The LFP source and destination ports can be physical or link aggregation ports. If the destination port is a link aggregation port the shutdown consists of shutting down all members of the link aggregation group (physically down). However, the link aggregation group remains administratively enabled.

Interaction With Interfaces Violation Recovery

- The **clear violation** command will clear the LFP violations and mark the interfaces as up even if the violation condition still exists.
- An admin down followed by an admin up will clear the LFP violation and mark the interfaces as up even if the violation condition still exists.
- When the destination port is a link aggregate, the shutdown action does not shutdown the link aggregation. Instead, all the ports that are members of the link aggregation at the time of the violation are shutdown.
- A link aggregate port remains in a violation state even if the port leaves the link aggregate.
- If a port that is not a member of a link aggregate at the time a violation occurred is added to a link aggregate, the switch will not shut down the port.
- SNMP traps cannot be configured for LFP. The interface violation recovery mechanism will be responsible for sending traps when a port is shutdown or recovered by LFP.
- If the wait-to-restore (WTR) timer is configured on the source ports of a LFP group with link monitoring enabled, the state of the destination ports of the group will be determined by the link state of the ports after the WTR timer has expired.

See “Interfaces Violation Recovery” on page 1-14 for information of learning port violations.

Configuring Link Fault Propagation

Configuring LFP requires the following steps:

1 Create an LFP group. This type of group identifies the source ports to monitor and the destination ports to bring down when all of the source ports go down. To create an LFP group, use the **link-fault-propagation group** command. For example:

```
-> link-fault-propagation group 1
```

2 Associate source ports with the LFP group. To associate source ports to an LFP group, use the **link-fault-propagation group source** command. For example:

```
-> link-fault-propagation group 1 source port 1/2-5 2/3
```

3 Associate destination ports with the LFP group. To associate destination ports with an LFP group, use the **link-fault-propagation group destination** command. For example:

```
-> link-fault-propagation group 1 destination port 1/5-8 2/3
```

4 Configure the LFP wait-to-shutdown timer. This timer specifies the amount of time that LFP will wait before shutting down all the destination ports. To configure this timer value, use the **link-fault-propagation group wait-to-shutdown** command. For example:

```
-> link-fault-propagation group 1 wait-to-shutdown 70
```

Note. *Optional*. To verify the LFP configuration, use the **show link-fault-propagation group** command. For example:

```
-> show link-fault-propagation group
Group Id : 2
  Source Port(s)          : 0/1-2 1/1-5 1/7,
  Destination Port(s)     : 0/3 1/10-13,
  Group-Src-Ports Status  : up,
  Admin Status            : enable,
  Wait To Shutdown        : 10

Group Id : 6
  Source Port(s)          : 1/2 1/6 1/9,
  Destination Port(s)     : 1/10-11 1/13,
  Group-Src-Ports Status  : down,
  Admin Status            : disable,
  Wait To Shutdown        : 5

-> show link-fault-propagation group 2
Group Id : 2
  Source Port(s)          : 0/1-2 1/1-5 1/7,
  Destination Port(s)     : 0/3 1/10-13,
  Group-Src-Ports Status  : up,
  Admin Status            : enable,
  Wait To Shutdown        : 10
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about LFP commands.

LFP Application Example

This section provides an example of using LFP in a layer 2 network configuration, as shown in the following sample topology:

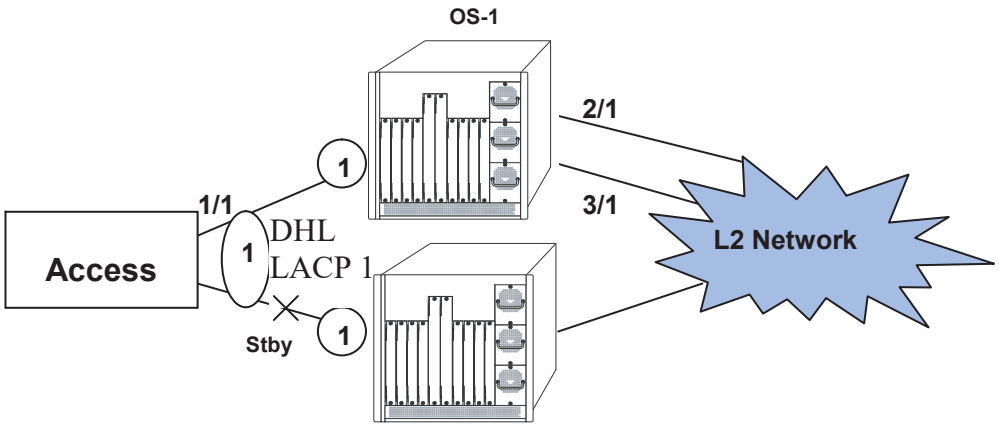


Figure 1-1 : Link Fault Propagation - Application Example

In this example:

- When interfaces 2/1 and 3/1 on OS-1 are down, the access switch will keep interface 1/1 as active and traffic will still be forwarded to OS-1 even though it has no network connectivity.
- To allow the switch to use the standby interface the link on OS-1 would need to be disabled so that interface 1/1 on the access switch leaves the LACP group.

```
-> link-fault-propagation group 1 source port 2/1 3/1 destination linkagg 1
```

IEEE 1588 Precision Time Protocol (PTP)

The Precision Time Protocol (PTP) is a protocol used to synchronize clocks throughout a computer network. On a local area network, it achieves clock accuracy in the sub-microsecond range, making it suitable for measurement and control systems.

The PTP function requires a time source device and a time receiver device to provide the time synchronization between the devices. The time source is called master and time receiver is known as slave. Apart from this master and slave clock devices, there are intermediate switches in the network through which the packets would be transmitted. On transmitting through these devices, a transmission delay would be introduced on each of the switches placed between the master and the slave. To maintain the accuracy of time synchronization between the master and the slave, it is important to take into consideration these transmit times taken by the PTP frames when passing along the intermediary nodes. The intermediate switches must have the ability to support PTP and thereby update the link and/or residency time of frames in these switches—a concept known as transparent clocking. There are two types of transparent clocks: end-to-end transparent clock and peer-to-peer transparent clock.

OmniSwitch supports both end-to-end transparent clock and one step peer-to-peer transparent clock.

Enabling/Disabling PTP Time Stamping

The **interfaces ptp admin-state** command can be used to enable or disable PTP on all the interfaces. PTP end-to-end transparent clock is supported in a standalone mode (virtual chassis of one) and virtual chassis of two.

```
-> interfaces ptp admin-state enable
-> interfaces ptp admin-state disable
```

The below command sets the internal priority for the incoming PTP packet as 4.

```
-> interfaces ptp admin-state enable priority 4
```

To set the internal priority to the default value 5, use the **default** keyword.

```
-> interfaces ptp admin-state enable priority default
```

To enable PTP end-to-end transparent clock in a virtual chassis of two, single loopback port per chassis is required to be configured. Use **interfaces ptp admin-state** command to configure the loopback port.

The loopback ports dedicated for PTP must not be used by any other feature. Ensure PTP is configured on unused ports. Also, PTP on a virtual chassis of two is not supported on chassis ID 1.

```
-> interfaces ptp admin-state enable loopback-portlist 2/1/12 3/1/23
WARNING: User ports 2/1/12 and 3/1/23 will be out of service for users.
```

Enabling/Disabling PTP Peer-to-Peer Transparent Clock

The **interfaces port ptp p2p** command can be used to enable or disable IEEE 1588 PTP peer-to-peer transparent clock on an interface. When peer-to-peer is enabled on a port, link delay will be computed dynamically for the corresponding link.

```
-> interfaces port 1/1/1 ptp p2p admin-state enable

-> interfaces port 1/1/1 ptp p2p admin-state disable
```

Notes:

- Ensure Loopback0 IP interface is configured on the switch as loopback0 interface address will be used as the source IP for peer delay measurement packets. If loopback0 IP interface is not configured, then peer delay measurement feature will not work.
- PTP must be enabled globally for PTP peer-to-peer support.
- PTP peer-to-peer supports only one-step mode.

MAC Security Overview

MACsec (MAC Security) provides point-to-point security on Ethernet links between directly connected nodes. MACsec prevents DoS/M-in-M/playback attacks, intrusion, wire-tapping, masquerading, and so on. MACsec can be used to secure traffic on Ethernet links - LLDP frames, LACP frames, DHCP/ARP packets, and so on.

MACsec feature requires a license. The MACsec license is a site license and does not use the serial number and MAC address of the switch. The MACsec license file can be applied using the 'license' command. For more information, refer to the ‘Chassis Management and Monitoring Commands’ chapter in *OmniSwitch AOS Release 8 CLI Reference Guide*.

How It Works?

MACsec-enabled links are secured by matching security keys. Data integrity checks are done by appending an 8-byte or 16-byte header and a 16-byte tail to all Ethernet frames traversing the secured link. Optionally, traffic can also be encrypted, if enabled by user configuration.

On the wire, a MACsec packet starts with an Ethernet header with etherType 0x88E5, followed by an 8-byte or 16-byte SecTag header containing information about the decryption key, a packet number and Secure Channel Identifier. The SecTag header is followed by the payload (which may be optionally encrypted), and the Integrity Check Value (ICV) generated by GCM-AES of size 16 bytes.

Each node in a MACsec-protected network has at least one transmit secure channel associated with a Secure Channel Identifier (SCI). Configuration parameters such as enable encryption or perform replay protection are stored in the context of the transmit secure channel. A single secure channel is unidirectional - that is, it can be applied to either inbound or outbound traffic.

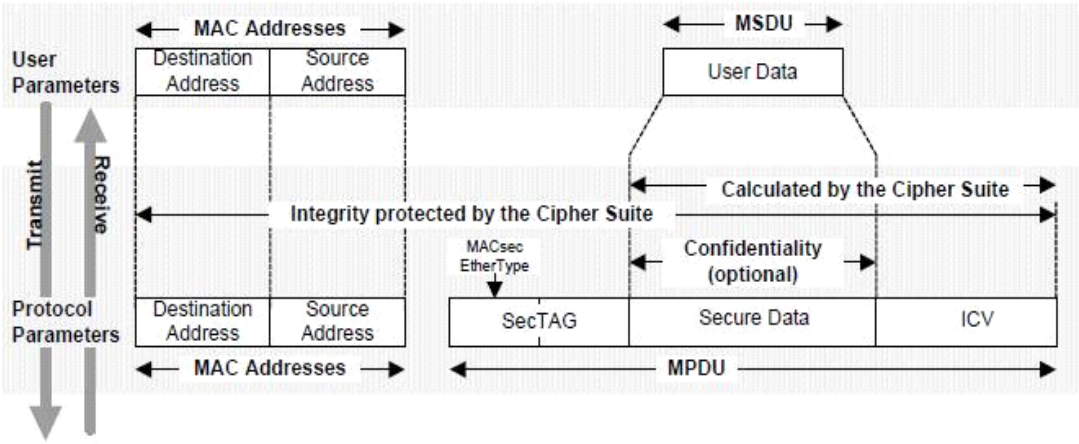


Figure 1-2 : MAC Security Overview

Each node that expects to receive traffic sent in a particular transmit secure channel must configure a 'matching' receive secure channel, with an SCI corresponding to the SCI of the transmit secure channel of the peer.

Within each secure channel, secure associations (SA) are defined. The SAs hold the encryption keys identified by their association number (AN), along with a packet number. On the transmit side, this packet number is put in the MACsec SecTag header and used in the encryption process. On the receive side, the packet number from the SecTag header will be checked against the packet number locally stored in the corresponding secure association to perform replay protection.

The default crypto suite used in MACsec is "128-bit AES-GCM" and the Session key is called a "Secure Association Key (SAK)". Each endpoint in a MACsec-protected network has at least one Tx Secure Channel (SCI-Tx) and multiple Rx Secure Channels (SCI-Rx). Between MACsec secure link, each endpoint point is configured with a matching SCI-Tx and SCI-Rx pair in both direction. Each Secure-Channel (SC) is associated with Secure Associations (SAs), which in turn holds the Secure Association Keys (SAK) along with a Packet Number (PN).

MACsec supports two SA modes:

- Static SA Mode - MACsec with Static Secure Association Key (static-SAK)
- Dynamic SA Mode - MACsec with Dynamic SAK using MACsec Key Agreement (MKA) Protocol. The MKA, as described in IEEE 802.1X-2010, is an extension to 802.1X, which provides the required session keys and manages the required encryption keys used by the underlying MACsec protocol. The MKA protocol allows peer discovery with confirmation of mutual authentication and sharing of MACsec secret keys to protect data exchanged by the peers.

  There are two modes of provisioning connectivity association keys (CAK/CKN) between two MACsec endpoints. OmniSwitch supports the following:

  – Dynamic SAK using Pre-Shared Key (PSK)
    MACsec using Static Connectivity Association Key (static-CAK) using PSK
  – Dynamic SAK using Extensible Authentication Protocol (EAP)
    MACsec using Dynamic Connectivity Association Key (dynamic-CAK) using EAP.

Static SA Mode

In static SA mode, manually configured SA keys are used to secure traffic on the point-to-point link between two nodes.

Consider the following configuration guidelines when MACsec is set to static SA mode:

- The SAK name and value are configured on SCI-Tx and SCI-Rx must be configured on both ends of the MACsec enabled interface.
- Each SAK name and value must have a corresponding matching value on the interface at the other end of the point-to-point Ethernet link to maintain MACsec on the link.
- Security is maintained by periodically rotating the SA keys between configured SAKs on SCI-Tx.

Dynamic SA Mode

In Dynamic SA Mode, Secure-Channel (SCI-Tx/SCI-Rx) and Secure-Association-Key (SAK) are exchanged between MACsec connected links using MKA protocol. The MKA protocol selects one of the nodes as the key server, which creates a dynamic SAK and shares it with the node at the other end over the secure channel. Once the other end also creates this dynamic SA key, subsequent traffic is secured using the new SA. The key server periodically and randomly creates and exchanges new SA to replace the older SA, using the MKA protocol for as long as the MACsec link is enabled.

Dynamic SAK using Pre-Shared Keys

This mode is applicable for securing link between two switch interfaces. Following are some configuration guidelines when MACsec is set to dynamic SA mode:

- Involves explicitly configuring a pre-shared key on a MACsec on both the ends of the point-to-point Ethernet link using a keychain, which triggers the MKA protocol to negotiate and generate necessary key for authentication and encryption.

- Two keys are used to secure the point-to-point Ethernet link.
  - A connectivity association key (CAK) that secures control plane traffic.
  - A randomly generated Secure Association Key (SAK) that secures data plane traffic.
- Both keys are exchanged between both the devices on each end of the point-to-point Ethernet link to ensure link security.
- The following action takes place while securing link between two switches using MKA.
  - The switches exchange MKPDUs verifying that the CAK/CKN pair match. A MACsec secured link is established using the user configured pre-shared key.
  - After the switches have mutually authenticated each other, one of the switches is elected as the key server.
  - Both the switches configure a pair of secure channels with matching identifiers.
  - Key server switch will then generate a key for each direction. These keys will be used to encrypt and decrypt the actual traffic.
  - Secure associations using these keys are configured on both the switches. **host and switch**.

Dynamic SAK using EAP

This mode is applicable for securing link between a host and a switch end-points. Following are some configuration guidelines when MACsec is set to dynamic SA mode using RADIUS server:

IEEE 802.1X-2010 defines the way that MACsec can be used in conjunction with authentication to provide secure port-based access control using authentication. IEEE 802.1X authenticates the endpoint and transmits the necessary cryptographic keying material to both sides. Using the master keys derived from the IEEE 802.1X authentication, MACsec can establish an encrypted link on the LAN, thereby helping ensure the security of the authenticated session.

- When configuring MACsec on a switch-to-host link, the MKA session establishment between the switch and the host is initiated once the 802.1x authentication is successful on the port. The 802.1x authentication method must be either EAP-TLS or PEAP authentication framework.
- The MKA keys are received from the RADIUS server. A successful 802.1x-authentication results in MKA keys (MSK and Session-Id), which will be passed from the RADIUS server to the switch and from RADIUS server to the host in an independent authentication transaction. The master key will then be passed between the switch and the host to create a MACsec secured connection. The CAK and CKN is derived from MSK and the EAP session ID.
- CAK and CKN needs to be derived both at the host and the switch, hence 802.1x-authentication using EAP-TLS must be used as mutual authentication protocol for MACsec Dynamic mode.

After deriving CAK/CKN, the switch acts as the key server. It generates a random SAK, which is sent to the client. The client is never a key server and can only interact with a single MKA entity, the key server. After key derivation and generation, the switch sends periodic transports to the client at a default interval of two seconds.

Key Management and Rotation

To support non-interrupting MACsec service, four keys are supported for each secure channel in MACsec Static Mode. One key is used for actively protecting the traffic, while the other keys are programmed into hardware to be used as backup. This would reduce the frequency that SW has to be interrupted to setup a new key. In MACsec Dynamic Mode, the key rotation would be handled in SW using packet number (PN) rollover using MKA protocol.

For more information on security key management commands, see the “Managing System Files” chapter in the *OmniSwitch AOS Release 8 Switch Management Guide*.

Enabling/Disabling MACsec on an Interface

Use **interfaces macsec admin-state** command to enable or disable MACsec on a physical port or a port range. Use this command to,

- Enable or disable MACsec.

Note. It is required to install the MACsec license to successfully enable MACsec.

- Set the MACsec mode: Static SA Mode or Dynamic SA Mode
- For Static SA Mode - following configurations can be configured.
  - Set the MACsec mode to ‘static’. By default, the MACsec mode is set to ‘static’.
  - Create MACsec Tx and Rx channels.
  - Specify the SCI value for Tx and Rx channels.
  - Associate the keychain ID for Tx and Rx channel. The keychain associated with the SCI-Tx and SCI-Rx must have four keys supporting ‘AES-GCM-128’ algorithm, and the number of keys in the keychain associated with both SCI-Tx and SCI-Rx on an interface must be equal.
  - Enable or disable encryption on Tx and Rx channel (optional).
- Dynamic SA Mode has two variations - Dynamic SAK using pre-shared keys and Dynamic SAK using Extensible Authentication Protocol (EAP).
  - For Dynamic SAK using pre-shared keys, following configurations can be configured.
    - Set the MACsec mode to ‘dynamic’.
    - Configure the keychain for Static-CAK. The keychain or pre-shared key for Static-CAK must have the key mapped either to ‘AES-CMAC-128’ algorithm or ‘AES-CMAC-256’ algorithm. AES-CMAC-256 option would be supported only on platforms supporting 256-bit key. View **show interfaces capability** output to check if the interfaces has the support for MACsec 256-bit encryption.
    - Configure key server priority (optional).
    - Configure transmit interval for MKPDUs (optional).
    - Enable or disable encryption on dynamic secure channel. (optional)
  - For Dynamic SAK using EAPs, the following configurations can be configured.
    - Set the MACsec mode to ‘radius’.
    - Configure transmit interval for MKPDUs (optional).
    - Enable or disable encryption on dynamic secure channel (optional).

For example, the following configures MACsec to static mode.

```
-> interface port 1/1/1 macsec admin-state enable mode static sci-tx 0x1 key-
chain 1 encryption sci-rx 0x1 key-chain 1 encryption
```

The following configures MACsec to dynamic using static CAK.

```
-> interface port 1/1/1 macsec admin-state enable mode dynamic key-chain 1
server-priority 10 transmit-interval 3
```

The following configures MACsec to dynamic SAK using EAP.

```
-> interface port 1/1/1 macsec admin-state enable mode dynamic radius
```

Use the **no** form of this command to disable encryption on Tx/Rx channel, remove keychain configuration on Tx/Rx channel, remove Tx/Rx channel. For example,

```
-> no interface 1/1/1 macsec sci-rx 0x2 keychain
-> no interface 1/1/1 macsec sci-tx encryption
```

Verifying the MACsec Configuration

To display the MACsec configuration on the switch, use the following **show** commands:

show interfaces macsec	Displays the MACsec configuration on a physical port or port range.
show interfaces macsec static	Displays the detailed MACsec configuration on a physical port or port range configured with MACsec mode 'Static.
show interfaces macsec dynamic	Displays the detailed MACsec configuration on a physical port or port range configured with MACsec mode 'Dynamic'.
show interfaces macsec statistics	Displays the MACsec statistics collected for a physical port.

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

2 Configuring UDLD

UniDirectional Link Detection (UDLD) is a protocol for detecting and disabling unidirectional Ethernet fiber or copper links caused by mis-wiring of fiber strands, interface malfunctions, media converter faults, and so on. The UDLD protocol operates at Layer 2 in conjunction with the IEEE 802.3 - Layer 1 fault detection mechanisms.

UDLD is a lightweight protocol that can be used to detect and disable one-way connections before they create dangerous situations such as Spanning Tree loops or other protocol malfunctions. The protocol is mainly used to advertise the identities of all the UDLD-capable devices attached to the same LAN segment and to collect the information received on the ports of each device to determine whether or not the Layer 2 communication is functioning properly. All connected devices must support UDLD for the protocol to successfully identify and disable unidirectional links. When UDLD detects a unidirectional link, the protocol administratively shuts down the affected port and generates a trap to alert the user.

In This Chapter

This chapter describes how to configure UDLD parameters through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include the following:

- "Configuring UDLD" on page 2-6.
- "Configuring the Operational Mode" on page 2-7.
- "Configuring the Probe-Timer" on page 2-7.
- "Configuring the Echo-Wait-Timer" on page 2-7.
- "Clearing UDLD Statistics" on page 2-8.
- "Verifying the UDLD Configuration" on page 2-8.
- "Verifying the UDLD Configuration" on page 2-8.

UDLD Defaults

Parameter Description	Command	Default
UDLD administrative state	**udld**	Disabled
UDLD status of a port	**udld port**	Disabled
UDLD operational mode	**udld mode**	Normal
Probe-message advertisement timer	**udld probe-timer**	15 seconds
Echo-based detection timer	**udld echo-wait-timer**	8 seconds

Quick Steps for Configuring UDLD

1 To enable the UDLD protocol on a switch, use the **udld** command. For example:

```
-> udld enable
```

2 To enable the UDLD protocol on a port, use the **udld port** command by entering **udld port**, followed by the slot and port number, and **enable**. For example:

```
-> udld port 1/6 enable
```

3 Configure the operational mode of UDLD by entering **udld port**, followed by the slot and port number, **mode**, and the operational mode. For example:

```
-> udld port 1/6 mode aggressive
```

4 Configure the probe-message advertisement timer on port 6 of slot 1 as 17 seconds using the following command:

```
-> udld port 1/6 probe-timer 17
```

Note. *Optional*. Verify the UDLD global configuration by entering the **show udld configuration** command or verify the UDLD configuration on a port by entering the **show udld configuration port** command. For example:

```
-> show udld configuration

Global UDLD Status : Disabled

-> show udld configuration port 1/6
Global UDLD Status: enabled
Port UDLD Status: enabled
Port UDLD State: bidirectional
UDLD Op-Mode: normal
Probe Timer (Sec): 20,
Echo-Wait Timer (Sec): 10
```

To verify the UDLD statistics of a port, use the **show udld statistics port** command. For example:

```
-> show udld statistics port 1/42

UDLD Port Statistics
 Hello Packet Send        :8,
 Echo Packet Send         :8,
 Flush Packet Recvd       :0
UDLD Neighbor Statistics
 Neighbor ID    Hello Pkts Recv       Echo Pkts Recv
--------------+--------------------+--------------
      1              8                15
      2              8                15
      3              8                21
      4              8                14
      5              8                15
      6              8                20
```

UDLD Overview

UDLD is a Layer 2 protocol used to examine the physical configuration connected through fiber-optic or twisted-pair Ethernet cables. When a port is affected and only a unidirectional link is working, UDLD detects and administratively shuts down the affected port, and alerts the user. Unidirectional links can create hazardous situations such as Spanning-Tree topology loops caused, for instance, by unwiring of fiber strands, interface malfunctions, faults of the media converter, and so on.

The UDLD feature is supported on the following port types:

- Copper ports
- Fiber ports

UDLD Operational Mode

UDLD supports two modes of operation:

- Normal mode
- Aggressive mode

UDLD works with the Layer 1 mechanisms to determine the physical status of a link. A unidirectional link occurs whenever the traffic sent from a local device is received by its neighbor; but the traffic from the neighbor is not received by the local device.

Normal Mode

In this mode, the protocol depends on explicit information instead of implicit information. If the protocol is unable to retrieve any explicit information, the port is not put in the shutdown state; instead, it is marked as **Undetermined**. The port is put in the shutdown state only when:

- It is explicitly determined that the link is defective
- When it is determined on the basis of UDLD-PDU processing that link has become unidirectional.

In any such state transition, a trap is raised.

Aggressive Mode

In this mode, UDLD checks whether the connections are correct and the traffic is flowing bidirectionally between the respective neighbors. The loss of communication with the neighbor is considered an event to put the port in shutdown state. Thus, if the UDLD PDUs are not received before the expiry of a timer, the port is put in the **UDLD-shutdown** state. Since the lack of information is not always due to a defective link, this mode is optional and is recommended only for point-to-point links.

UDLD shuts down the affected interface when one of these problems occurs:

- On fiber-optic or twisted-pair links, one of the interfaces cannot send or receive traffic.
- On fiber-optic or twisted-pair links, one of the interfaces is down while the other is up.
- One of the fiber strands in the cable is disconnected.

Mechanisms to Detect Unidirectional Links

The UDLD protocol is implemented to correct certain assumptions made by other protocols and to help the Spanning Tree Protocol to function properly to avoid dangerous Layer 2 loops.

UDLD uses two basic mechanisms:

- It advertises the identity of a port and learns about its neighbors. This information about the neighbors is maintained in a cache table.
- It sends continuous echo messages in certain circumstances that require fast notifications or fast re-synchronization of the cached information.

Neighbor database maintenance

UDLD learns about other UDLD neighbors by periodically sending a Hello packet (also called an advertisement or probe) on every active interface to inform each device about its neighbors.

When the switch receives a Hello message, the switch caches the information until the age time expires. If the switch receives a new Hello message before the aging of an older cache entry, the switch replaces the older entry with the new one.

Whenever an interface is disabled and UDLD is running, or UDLD is disabled on an interface, or the switch is reset, UDLD clears all the existing cache entries for the interfaces that are affected by the configuration change. UDLD sends a message to the neighbors to flush the part of their caches affected by the status change. This UDLD message is intended to synchronize the caches.

Echo detection

UDLD depends on an echo-detection mechanism. UDLD restarts the detection window on its side of the connection and sends echo messages in response to the request, whenever a UDLD device learns about a new neighbor or receives a re-synchronization request from an out-of-sync neighbor. This behavior is the same on all UDLD neighbors because the sender of the echoes expects to receive an echo as a response.

If the detection window ends and no valid response is received, the link is shut down, depending on the UDLD mode. When UDLD is in normal mode, the link is considered to be undetermined and is not shut down. When UDLD is in aggressive mode, the link is considered to be unidirectional, and the interface is shut down.

In normal mode, if UDLD is in the advertisement or in the detection phase and all the neighbor cache entries are aged out, UDLD restarts the link-up sequence to re-synchronize with potentially out-of-sync neighbors.

In aggressive mode, if UDLD is in the advertisement or in the detection phase and all the neighbors of a port are aged out, UDLD restarts the link-up sequence to re-synchronize with potentially out-of-sync neighbors. UDLD shuts down the port, after the continuous messages, if the link state is undetermined.

Configuring UDLD

This section describes how to use Command Line Interface (CLI) commands to do the following:

- "Enabling and Disabling UDLD" on page 2-6.
- "Configuring the Operational Mode" on page 2-7.
- "Configuring the Probe-Timer" on page 2-7.
- "Configuring the Echo-Wait-Timer" on page 2-7.
- "Clearing UDLD Statistics" on page 2-8.
- "Verifying the UDLD Configuration" on page 2-8.

Note. See the "UDLD Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for complete documentation of UDLD CLI commands.

Enabling and Disabling UDLD

By default, UDLD is disabled on all switch ports. To enable UDLD on a switch, use the **udld** command. For example, the following command enables UDLD on a switch:

```
-> udld enable
```

To disable UDLD on a switch, use the **udld** command with the **disable** parameter. For example, the following command disables UDLD on a switch:

```
-> udld disable
```

Enabling UDLD on a Port

By default, UDLD is disabled on all switch ports. To enable UDLD on a port, use the **udld port** command. For example, the following command enables UDLD on port 3 of slot 1:

```
-> udld port 1/3 enable
```

To enable UDLD on multiple ports, specify a range of ports. For example:

```
-> udld port 1/6-10 enable
```

To disable UDLD on a port, use the **udld port** command with the **disable** parameter. For example, the following command disables UDLD on a range of ports:

```
-> udld port 5/21-24 disable
```

Configuring the Operational Mode

To configure the operational mode, use the **udld mode** command as shown:

```
-> udld mode aggressive
```

For example, to configure the mode for port 4 on slot 2, enter:

```
-> udld port 2/4 mode aggressive
```

To configure the mode for multiple ports, specify a range of ports. For example:

```
-> udld port 2/7-18 mode normal
```

Configuring the Probe-Timer

To configure the probe-message advertisement timer, use the **udld probe-timer** command as shown:

```
-> udld probe-timer 20
```

For example, to configure the probe-timer for port 3 on slot 6, enter:

```
-> udld port 6/3 probe-timer 18
```

To configure the probe-timer for multiple ports, specify a range of ports. For example:

```
-> udld port 1/8-21 probe-timer 18
```

Use the **no** form of this command to reset the timer. For example, the following command resets the timer for port 4 of slot 6:

```
-> no udld port 6/4 probe-timer
```

The following command resets the timer for multiple ports:

```
-> no udld port 1/8-21 probe-timer
```

Configuring the Echo-Wait-Timer

To configure the echo-based detection timer, use the **udld echo-wait-timer** command as shown:

```
-> udld echo-wait-timer 9
```

For example, to configure the echo-wait-timer for port 5 on slot 6, enter:

```
-> udld port 6/5 echo-wait-timer 12
```

To configure the echo-wait-timer for multiple ports, specify a range of ports. For example:

```
-> udld port 1/8-21 echo-wait-timer 9
```

Use the **no** form of this command to reset the timer. For example, the following command resets the timer for port 6 of slot 4:

```
-> no udld port 4/6 echo-wait-timer
```

The following command resets the timer for multiple ports:

```
-> no udld port 1/8-21 echo-wait-timer
```

Clearing UDLD Statistics

To clear the UDLD statistics, use the **clear udld statistics port** command. For example, to clear the statistics for port 4 on slot 1, enter:

```
-> clear udld statistics port 1/4
```

To clear the UDLD statistics on all the ports, enter:

```
-> clear udld statistics
```

Verifying the UDLD Configuration

To display UDLD configuration and statistics information, use the show commands listed below:

show udld configuration	Displays the global status of UDLD configuration.
show udld configuration port	Displays the configuration information for all UDLD ports or for a particular UDLD port on the switch.
show udld statistics port	Displays the UDLD statistics for a specific port.
show udld neighbor port	Displays the UDLD neighbor ports.
show udld status port	Displays the UDLD status for all ports or for a specific port.

For more information about the resulting display from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. An example of the output for the **show udld configuration port** and **show udld statistics port** commands is also given in "Quick Steps for Configuring UDLD" on page 2-3.

3 Managing Source Learning

Transparent bridging relies on a process referred to as *source learning* to handle traffic flow. Network devices communicate by sending and receiving data packets that each contain a source MAC address and a destination MAC address. When packets are received on switch network interface (NI) module ports, source learning examines each packet and compares the source MAC address to entries in a MAC address database table. If the table does not contain an entry for the source address, then a new record is created associating the address with the port it was learned on. If an entry for the source address already exists in the table, a new one is not created.

Packets are also filtered to determine if the source and destination address are on the same LAN segment. If the destination address is not found in the MAC address table, then the packet is forwarded to all other switches that are connected to the same LAN. If the MAC address table does contain a matching entry for the destination address, then there is no need to forward the packet to the rest of the network.

In This Chapter

This chapter describes how to manage source learning entries in the switch MAC address table (often referred to as the *forwarding* or *filtering database*) through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- "Using Static MAC Addresses" on page 3-3.
- "Using Static Multicast MAC Addresses" on page 3-5.
- "Configuring MAC Address Table Aging Time" on page 3-7.
- "Configuring the Source Learning Status" on page 3-8.
- "Increasing the MAC Address Table Size" on page 3-9.
- "Displaying Source Learning Information" on page 3-10.

Source Learning Defaults

Parameter Description	Command	Default
Static MAC address operating mode	**mac-learning static mac-address**	bridging
MAC address aging timer	**mac-learning aging-time**	300 seconds
MAC source learning status per port	**mac-learning**	enabled
MAC source learning mode	**mac-learning mode**	centralized

MAC Address Table Overview

Source learning builds and maintains the MAC address table on each switch. New MAC address table entries are created in one of two ways: they are dynamically learned or statically assigned. Dynamically learned MAC addresses are those that are obtained by the switch when source learning examines data packets and records the source address and the port and VLAN it was learned on. Static MAC addresses are user defined addresses that are statically assigned to a port and VLAN using the **mac-learning static mac-address** command.

Accessing MAC Address Table entries is useful for managing traffic flow and troubleshooting network device connectivity problems. For example, if a workstation connected to the switch is unable to communicate with another workstation connected to the same switch, the MAC address table might show that one of these devices was learned on a port that belonged to a different VLAN or the source MAC address of one of the devices do not appear at all in the address table.

Using Static MAC Addresses

Static MAC addresses are configured using the **mac-learning static mac-address** command. These addresses direct network traffic to a specific port and VLAN. They are particularly useful when dealing with silent network devices. These types of devices do not send packets, so their source MAC address is never learned and recorded in the MAC address table. Assigning a MAC address to the silent device's port creates a record in the MAC address table and ensures that packets destined for the silent device are forwarded out that port.

When defining a static MAC address for a particular slot/port and VLAN, consider the following:

- Configuring static MAC addresses is only supported on fixed ports.
- The specified slot/port must already belong to the specified VLAN. Use the **vlan members untagged** command to assign a port to a VLAN before you configure the static MAC address.
- Only traffic from other ports associated with the same VLAN is directed to the static MAC address slot/port.
- Static MAC addresses are **permanent** addresses. This means that a static MAC address remains in use even if the MAC ages out or the switch is rebooted.
- There are two types of static MAC address behavior supported: **bridging** (default) or **filtering**. Enter **filtering** to set up a denial of service to block potential hostile attacks. Traffic sent to or from a filtered MAC address is dropped. Enter **bridging** for regular traffic flow to or from the MAC address.
- If a packet received on a port associated with the same VLAN contains a source address that matches a static MAC address, the packet is discarded. The same source address on different ports within the same VLAN is not supported.
- If a static MAC address is configured on a port link that is down or disabled, an asterisk appears to the right of the MAC address in the display output. The asterisk indicates that this is an invalid MAC address. When the port link comes up, however, the MAC address is then considered valid and the asterisk no longer appears next to the address in the display.

Configuring Static MAC Addresses

To configure a permanent, bridging static MAC address, see the example below:

```
-> mac-learning vlan 1 port 1/1 static mac-address 00:00:02:CE:10:37 bridging
```

Use the **no** form of this command to clear MAC address entries from the table:

```
-> no mac-learning vlan 1 port 1/1 static mac-address 00:00:02:CE:10:37 bridging
```

To verify static MAC address configuration and other table entries, use the **show mac-learning** command. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Static MAC Addresses on Link Aggregate Ports

Static MAC Addresses are not assigned to physical ports that belong to a link aggregate. Instead, they are assigned to a link aggregate ID that represents a collection of physical ports. This ID is specified at the time the link aggregate of ports is created.

To configure a static MAC address on a link aggregate ID 1 belong to VLAN 1 see the example below:

```
-> mac-learning vlan 1 linkagg 1 static mac-address 00:00:02:CE:10:37 bridging
```

For more information about configuring a link aggregate of ports, see Chapter 9, “Configuring Static Link Aggregation” and Chapter 10, “Configuring Dynamic Link Aggregation.”

Using Static Multicast MAC Addresses

Using static multicast MAC addresses allows you to send traffic intended for a single destination multicast MAC address to selected switch ports within a given VLAN. To specify which ports receive the multicast traffic, a static multicast address is assigned to each selected port for a given VLAN. The ports associated with the multicast address are then identified as egress ports. When traffic received on ports within the same VLAN is destined for the multicast address, the traffic is forwarded only on the egress ports that are associated with the multicast address.

The **mac-learning multicast mac-address** command is used to configure a static multicast MAC address. When defining this type of static MAC address for a particular port and VLAN, consider the following:

- A MAC address is considered a multicast MAC address if the least significant bit of the most significant octet of the address is enabled. For example, MAC addresses with a prefix of 01, 03, 05, 13, etc., are multicast MAC addresses.
- If a multicast prefix value is not present, then the address is treated as a regular MAC address and not allowed with the **mac-learning vlan multicast mac-address** command.
- The multicast addresses within the following ranges are not supported:

  01:00:5E:00:00:00 to 01:00:5E:7F:FF:FF
  01:80:C2:XX.XX.XX
  33:33:XX:XX:XX:XX

- In addition to configuring the same static multicast address for multiple ports within a given VLAN, it is also possible to use the same multicast address across multiple VLANs.
- The specified port or link aggregate ID must already belong to the specified VLAN.

Configuring Static Multicast MAC Addresses

The **mac-learning multicast mac-address** command is used to define a destination multicast MAC address and assign the address to one or more egress ports within a specified VLAN. For example, the following command assigns the multicast address 01:25:9a:5c:2f:10 to port 1/24 in VLAN 20:

```
-> mac-learning vlan 20 port 1/1 multicast mac-address 01:25:9a:5c:2f:10
```

Use the **no** form of the **mac-learning multicast mac-address** command to delete static multicast MAC address entries:

```
-> no mac-learning vlan 20 port 1/1 multicast mac-address 01:25:9a:5c:2f:10
```

If a MAC address, slot/port and VLAN ID are not specified with this form of the command, then all static multicast addresses are deleted. For example, the following command deletes all static MAC addresses, regardless of their slot/port or VLAN assignments:

```
-> no mac-learning multicast
```

To verify the static MAC address configuration and other table entries, use the **show mac-learning** and **show mac-learning** commands. For more information about these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Static Multicast MAC Addresses on Link Aggregate Ports

Static multicast MAC addresses are not assigned to physical ports that belong to a link aggregate. Instead, they are assigned to a link aggregate ID that represents a collection of physical ports. This ID is specified at the time the link aggregate of ports is created and when using the **mac-address-table static-multicast** command.

To configure a static multicast MAC address on a link aggregate ID, use the **mac-learning multicast mac-address** command with the **linkagg** keyword to specify the link aggregate ID. For example, the following command assigns a static multicast MAC address to link aggregate ID 2 associated with VLAN 455:

```
-> mac-learning vlan 455 linkagg 2 multicast mac-address 01:95:2A:00:3E:4c
```

Configuring MAC Address Table Aging Time

Source learning also tracks MAC address age and removes addresses from the MAC address table that have aged beyond the aging timer value. When a device stops sending packets, source learning keeps track of how much time has passed since the last packet was received on the switch port of the device. When this amount of time exceeds the aging time value, the MAC is *aged out* of the MAC address table. Source learning always starts tracking MAC address age from the time since the last packet was received.

For example, the following sets the aging time for all VLANs to 1200 seconds (20 minutes):

```
-> mac-learning aging-time 1200
```

A MAC address learned on any VLAN port ages out when the time since a packet with the particular address was last seen on the port exceeds 1200 seconds.

Note. An inactive MAC address can take up to twice as long as the aging time value specified to age out of the MAC address table. For example, if an aging time of 60 seconds is specified, the MAC ages out any time between 60 and 120 seconds of inactivity.

To set the aging time back to the default value, use the **default** parameter. For example, the following sets the aging time for all VLANs back to the default value:

```
-> mac-learning aging-time default
```

To display the aging time value use the **show mac-learning aging-time** command. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring the Source Learning Status

The source learning status for a port or link aggregate of ports is configurable using the **mac-learning** command. For example:

```
-> mac-learning port 1/10 disable
-> mac-learning port 1/15-20 disable
-> mac-learning linkagg 10 disable
```

To enable the source learning status for a port or link aggregate, use the **source-learning** command with the **enable** option. For example:

```
-> mac-learning port 1/10 enable
-> mac-learning port 1/15-20 enable
-> mac-learning linkagg 10 enable
```

Disabling source learning on a port or link aggregate is useful on a ring configuration, where a switch within the ring does not need to learn the MAC addresses that the same switch is forwarding to another switch within the ring,. This functionality is also useful in Transparent LAN Service configurations, where the service provider device does not need to learn the MAC addresses of the customer network.

Configuring the source learning status is not allowed on the following types of switch ports:

- Ports enabled with Learned Port Security (LPS).
- Ports enabled with Universal Network Profile (UNP) functionality.
- Member ports of a link aggregate.

Consider the following guidelines when changing the source learning status for a port or link aggregate:

- Disabling source learning on a link aggregate disables MAC address learning on all member ports of the link aggregate.
- MAC addresses dynamically learned on a port or aggregate are cleared when source learning is disabled.
- Statically configured MAC addresses are not cleared when source learning is disabled for the port or aggregate. In addition, configuring a new static MAC address is allowed even when source learning is disabled.

Increasing the MAC Address Table Size

There are two source learning modes available for the OmniSwitch: centralized and distributed. Enabling the distributed mode for the switch increases the table size for the switch.

To enable the distributed MAC source learning mode for the chassis, use the **mac-learning mode** command. When this mode is disabled, the switch operates in the centralized MAC source learning mode (the default).

Enabling or disabling the distributed MAC source learning mode requires the following three steps:

1. Set the mode.
2. Enter the **write memory** command to save the switch configuration.
3. Reboot the switch.

For example:

```
-> mac-learning mode distributed

WARNING: Source Learning mode has changed - must do write memory and reload

-> write memory

-> reload
```

Note. All three of the above configuration steps are required to enable or disable the MAC mode. If any of the above steps are skipped, the status of the mode is not changed.

Displaying Source Learning Information

To display MAC Address Table entries, statistics, and aging time values, use the show commands listed below:

show mac-learning	Displays a list of all MAC addresses known to the MAC address table, including static and multicast MAC addresses.
show mac-learning aging-time	Displays the current MAC address aging timer value by switch or VLAN.
show mac-learning mode	Displays the current status of the distributed MAC source learning mode.

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

4 Configuring VLANs

In a flat bridged network, a broadcast domain is confined to a single LAN segment or even a specific physical location, such as a department or building floor. In a switch-based network, such as one comprised of OmniSwitch systems, a broadcast domain, or VLAN can span multiple physical switches and can include ports from a variety of media types. For example, a single VLAN could span three different switches located in different buildings and include a variety of Ethernet port configurations, such as 802.1q tagged VLAN member ports and/or a link aggregate of ports.

In This Chapter

This chapter describes how to define and manage VLAN configurations through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- "Creating/Modifying VLANs" on page 4-4.
- "Assigning Ports to VLANs" on page 4-6.
- "Enabling/Disabling Spanning Tree for a VLAN" on page 4-9.
- "Enabling/Disabling Source Learning" on page 4-9.
- "Configuring VLAN IP Interfaces" on page 4-10.
- "Bridging VLANs Across Multiple Switches" on page 4-11.
- "Verifying the VLAN Configuration" on page 4-13.
- "Using Private VLANs" on page 4-15.

For information about Spanning Tree, see Chapter 6, "Configuring Spanning Tree Parameters."

For information about routing, see Chapter 17, "Configuring IP."

VLAN Defaults

Parameter Description	Command	Default
VLAN identifier (VLAN ID)	**vlan**	VLAN 1 predefined on each switch.
VLAN administrative state	**vlan**	Enabled
VLAN description	**vlan name**	VLAN ID
VLAN Spanning Tree state	**spantree vlan admin-state**	Enabled
VLAN IP router interface	**ip interface**	None
VLAN port associations	**vlan members untagged**	All ports are initially associated with default VLAN 1.

Sample VLAN Configuration

The following steps provide a quick tutorial to create VLAN 100. Also included are steps to define a VLAN description, IP router interface, and static switch port assignments.

Note. *Optional.* Creating a new VLAN involves specifying a VLAN ID that is not already assigned to an existing VLAN. To determine if a VLAN already exists in the switch configuration, enter **show vlan**. If VLAN 100 does not appear in the **show vlan** output, then it does not exist on the switch. For example:

```
-> show vlan
vlan  type  admin  oper  ip   mtu   name
-----+------+-----+-----+-----+-----+------------
   1   std    Ena   Dis   Dis  1500   VLAN 1
```

---

1 Create VLAN 100 with a description (for example, Finance IP Network) using the following command:

```
-> vlan 100 name "Finance IP Network"
```

2 Define an IP interface using the following command to assign an IP host address of 21.0.0.10 to VLAN 100 that enables forwarding of VLAN traffic to other subnets:

```
-> ip interface vlan_100_ip address 21.0.0.10 vlan 100
```

3 Assign switch ports 2 through 4 on slot 3 to VLAN 100 using the following command:

```
-> vlan 100 members port 3/2-4 untagged
```

Note. *Optional*. To verify the VLAN 100 configuration, use the **show vlan** command. For example:

```
-> show vlan 100
Name                     : Finance IP Network,
Type                     : Static Vlan,
Administrative State     : Enabled,
Operational State        : Disabled,
IP Router Port           : 21.0.0.10  255.0.0.0  forward  e2,
IP MTU                   : 1500
```

To verify that ports 3/2-4 were assigned to VLAN 100, use the **show vlan members** command. For example:

```
-> show vlan 100 members
  port     type      status
--------+---------+--------------
   3/2    default   inactive
   3/3    default   inactive
   3/4    default   inactive
```

To verify the details about the specific VLAN port 3/2, use the **show vlan members** command with the **port** keyword and port number. For example:

```
-> show vlan 100 members port 3/2
type       : default,
status     : inactive,
vlan admin : disabled,
vlan oper  : disabled,
```

VLAN Management Overview

One of the main benefits of using VLANs to segment network traffic, is that VLAN configuration and port assignment is handled through switch software. This eliminates the need to physically change a network device connection or location when adding or removing devices from the VLAN broadcast domain. The OmniSwitch VLAN management software handles the following VLAN configuration tasks:

- Creating or modifying VLANs.
- Assigning or changing default VLAN port associations (VPAs).
- Enabling or disabling VLAN participation in the current Spanning Tree algorithm.
- Displaying VLAN configuration information.

In addition to the above tasks, VLAN management software tracks and reports the following information to other switch software applications:

- VLAN configuration changes, such as adding or deleting VLANs, modifying the status of VLAN properties (for example, administrative, Spanning Tree, and authentication status), changing the VLAN description, or configuring VLAN router interfaces.
- VLAN port associations triggered by VLAN management and other switch software applications, such as 802.1Q VLAN tagging.
- The VLAN operational state, which is inactive until at least one active switch port is associated with the VLAN.

Creating/Modifying VLANs

The initial configuration for all OmniSwitch consists of a default VLAN 1 and all switch ports are initially assigned to this VLAN. When a switching module is added to the switch, the physical ports are also assigned to VLAN 1. If additional VLANs are not configured on the switch, then the entire switch is treated as one large broadcast domain. All ports receive traffic from all other ports.

In compliance with the IEEE 802.1Q standard, each VLAN is identified by a unique number, referred to as the “VLAN ID”. The user specifies a VLAN ID to create, modify or remove a VLAN and to assign switch ports to a VLAN. When a packet is received on a port, the VLAN ID of the port is inserted into the packet. The packet is then bridged to other ports that are assigned to the same VLAN ID. In essence, the VLAN broadcast domain is defined by a collection of ports and packets assigned to its VLAN ID.

The operational status of a VLAN remains inactive until at least one active switch port is assigned to the VLAN. This means that VLAN properties, such as Spanning Tree or router interfaces, also remain inactive. Ports are considered active if they are connected to an active network device. Non-active port assignments are allowed, but do not change the operational state of the VLAN.

Ports can be statically assigned to VLANs. When a port is assigned to a VLAN, a VLAN port association (VPA) is created and tracked by VLAN management switch software. For more information about VPAs, see “Assigning Ports to VLANs” on page 4-6.

Adding/Removing a VLAN

To add a VLAN to the switch configuration, enter **vlan** followed by a unique VLAN ID, an optional administrative status, and an optional description. For example, the following command creates VLAN 755 with a description:

```
-> vlan 755 name "IP Finance Network"
```

By default, administrative status and Spanning Tree are enabled when the VLAN is created. The **name** parameter for a VLAN is optional.

Note. Quotation marks are required if the description contains multiple words separated by spaces. If the description consists of only one word or multiple words separated by another character, such as a hyphen, then quotes are not required.

You can also specify a contiguous range of VLAN IDs by using a hyphen with the **vlan** command. For example, the following commands create VLANs 10 through 15 and 100 through 105 on the switch:

```
-> vlan 10-15 name "Marketing Network"
-> vlan 100-105 name "Marketing Network"
```

To remove a VLAN from the switch configuration, use the **no** form of the **vlan** command.

```
-> no vlan 200
-> no vlan 100-105
-> no vlan 10-15
```

When a VLAN is deleted, any router interfaces defined for the VLAN are removed and all VLAN port associations are dropped. If the VLAN deleted is the default VLAN for a port, the port returns to default VLAN 1. If the VLAN deleted is not a default VLAN, then the ports are directly detached from the VLAN. For more information about VLAN router interfaces, see "Configuring VLAN IP Interfaces" on page 4-10.

To view a list of VLANs already configured on the switch, use the **show vlan** command. See "Verifying the VLAN Configuration" on page 4-13 for more information.

Enabling/Disabling the VLAN Administrative Status

To enable or disable the administrative status for an existing VLAN, enter **vlan** followed by an existing VLAN ID and either **enable** or **disable**.

```
-> vlan 7 admin-state disable
-> vlan 1 admin-state enable
```

When the administrative status for a VLAN is disabled, VLAN port assignments are retained but traffic is not forwarded on these ports.

Modifying the VLAN Description

To change the description for a VLAN, enter **vlan** followed by an existing VLAN ID and the keyword **name** followed by the new description. For example, the following command changes the description for VLAN 455 to "Marketing IP Network":

```
-> vlan 455 name "Marketing IP Network"
```

Assigning Ports to VLANs

The OmniSwitch supports static assignment of physical switch ports to a VLAN. Once the assignment occurs, a VLAN port association (VPA) is created and tracked by VLAN management software on each switch. To view current VLAN port assignments in the switch configuration, use the **show vlan members** command.

Methods for statically assigning ports to VLANs include the following:

- Using the **vlan members untagged** command to define a new configured default VLAN for fixed ports. See "Changing the Default VLAN Assignment for a Port" on page 4-6.
- Using the **vlan members tagged** command to define 802.1Q-tagged VLANs for fixed ports. This method allows the switch to bridge traffic for multiple VLANs over one physical port connection. See "Using 802.1Q Tagging" on page 4-7.
- Configuring ports as members of a link aggregate that is assigned to a configured default VLAN. (See Chapter 9, "Configuring Static Link Aggregation," for more information.)

Changing the Default VLAN Assignment for a Port

Initially all switch ports are assigned to VLAN 1, which is also their *configured default* VLAN. When additional VLANs are created on the switch, ports are assigned to the VLANs so that traffic from devices connected to these ports is bridged within the VLAN domain.

NNI interface can also be configured with normal VLANs (other than VLAN 1) using this command. This allows NNI ports to carry untagged packets for the specified VLAN ID.

To assign a switch port to a new default VLAN, use the **vlan members untagged** command. For example, the following command assigns port 5 on slot 2 to VLAN 955:

```
-> vlan 955 members port 2/5 untagged
```

When the **vlan members** command is used, the port's default VLAN assignment is changed to the specified VLAN. The previous default VLAN assignment for the port (for example, VLAN 1, VLAN 10 or VLAN 200) is dropped.

The **vlan members** command is also used to change the default VLAN assignment for an aggregate of ports. The link aggregate control number is specified instead of a slot and port. For example, the following command assigns link aggregate 10 to VLAN 755:

```
-> vlan 755 members linkagg 10 untagged
```

For more information about configuring an aggregate of ports, see Chapter 9, "Configuring Static Link Aggregation."

Use the **no** form of the **vlan members** command to remove a default VPA. When this is done, VLAN 1 is restored as the default VLAN for the port.

```
-> no vlan 955 members port 2/5
```

Using 802.1Q Tagging

Another method for assigning ports to VLANs involves configuring a switch port or link aggregate to process 802.1Q-tagged frames that contain a specific VLAN ID designation. This method, referred to as 802.1Q tagging (or trunking), allows a single network link to carry traffic for multiple VLANs.

The OmniSwitch implements the IEEE 802.1Q standard for sending frames through the network tagged with VLAN identification. This section details procedures for configuring and monitoring 802.1Q tagging on a single switch port or link aggregate group.

"Tagged" refers to four bytes of reserved space in the header of the packet. The four bytes of "tagging" are broken down as follows: the first two bytes indicate whether the packet is an 802.1Q packet, and the next two bytes carry the VLAN identification (VID) and priority.

When packets ingress the switch, they are classified into a VLAN based on their 802.1Q tag information.

- If the packet contains an 802.1Q tag, the VLAN ID in the tag must match either the default VLAN ID for the port or a VLAN ID for which the port is tagged. If there is no match, the packet is dropped.
- If the packet is not tagged at all, the packet is placed into the default VLAN to which the port that received the packet is assigned.

The following diagram illustrates a simple network by using tagged and untagged traffic:

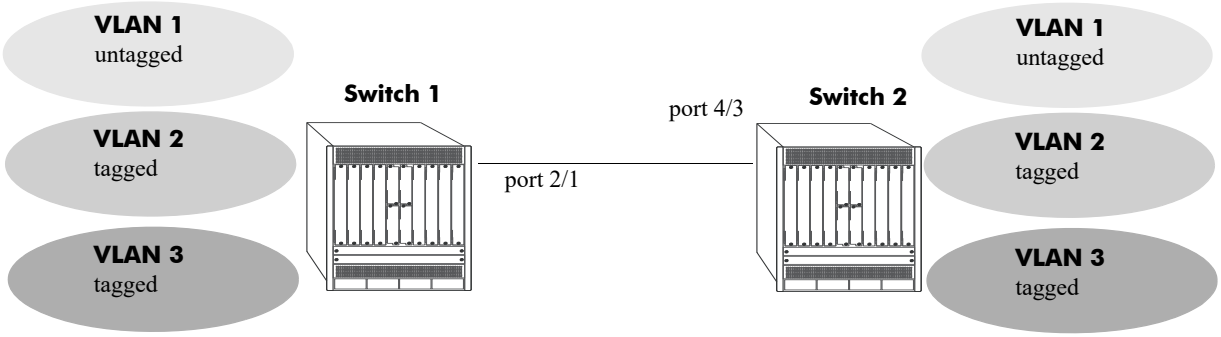


Figure 4-1 :Tagged and Untagged Traffic Network

Switch 1 and 2 have three VLANs, one for untagged traffic and two for tagged traffic. The ports connecting Switch 1 and 2 are configured in such a manner that the ports accept both tagged traffic for VLANS 2 and 3 and untagged traffic for VLAN 1.

A port can only be assigned to one untagged VLAN (in every case, this is the default VLAN configuration). In this example the default VLAN for port 2/1 and port 4/3 is VLAN 1. The port can be assigned to as many 802.1Q-tagged VLANs as necessary.

Configuring 802.1Q Tagging

To set a port to be a tagged port, use the **vlan members tagged** command and specify a VLAN identification (VID) number and a port number.

NNI ports can also be tagged with normal VLANs using this command. This allows NNI ports to carry regular 802.1q tagged traffic as well as SVLAN tagged traffic. 802.1q VLAN tagging to an NNI interface will not be allowed if the interface is set with TPID other than 0x8100.

For example, to configure port 3/4 to carry traffic for VLAN 5, enter the following command at the CLI prompt:

```
-> vlan 5 members port 4/3 tagged
```

Port 4/3 is now configured to carry packets tagged with VLAN 5, even though VLAN 5 is not the default VLAN for the port.

To enable tagging on link aggregation groups, enter the link aggregation group identification number in place of the slot and port number, as shown:

```
-> vlan 5 members linkagg 8 tagged
```

(For further information on creating link aggregation groups, see Chapter 9, "Configuring Static Link Aggregation," or Chapter 10, "Configuring Dynamic Link Aggregation.")

To remove 802.1Q tagging from a selected port or link aggregate, use the **untagged** parameter.

```
-> vlan 5 members linkagg 8 untagged
```

To display all VLANs, enter the following command:

```
-> show vlan port
```

Note. The link aggregation group must be created first before it can be set to use 802.1Q tagging.

Enabling/Disabling Spanning Tree for a VLAN

The Spanning Tree operating mode for the switch determines how VLAN ports are evaluated to identify redundant data paths. If the Spanning Tree switch operating mode is set to *flat*, then VLAN port connections are checked against other VLAN port connections for redundant data paths.

Note. The single flat mode STP instance is referred to as the CIST (Common and Internal Spanning Tree) instance.

In the flat mode, if the CIST instance is disabled, then it is disabled for all configured VLANs. However, disabling STP on an individual VLAN excludes only those VLAN ports from the flat STP algorithm.

If the Spanning Tree operating mode is set to *per-vlan* mode, there is a single Spanning Tree instance for each VLAN broadcast domain. Enabling or disabling STP on a VLAN in this mode includes or excludes the VLAN from the per-vlan STP algorithm.

The **spantree vlan admin-state** command is used to enable or disable a Spanning Tree instance for an existing VLAN. In the following examples, Spanning Tree is disabled on VLAN 255 and enabled on VLAN 755:

```
-> spantree vlan 255 admin-state disable
-> spantree vlan 755 admin-state enable
```

STP does not become operationally active on a VLAN unless the VLAN is operationally active, which occurs when at least one active port is assigned to the VLAN. Also, STP is enabled/disabled on individual ports. So even if STP is enabled for the VLAN, a port assigned to that VLAN must also have STP enabled. See Chapter 6, “Configuring Spanning Tree Parameters.”

Enabling/Disabling Source Learning

Source learning can be disabled on a VLAN. Disabling source learning can be beneficial in a ring topology. There is no limit on the number of ports that can belong to a VLAN that has source learning disabled, but it is recommended to include only the two ports connecting the switch to a ring.

To enable or disable source learning on a VLAN, use the **mac-learning static mac-address** command. For example, the following commands disable and enabled source learning on VLAN 10:

```
-> mac-learning vlan 10 disable
-> mac-learning vlan 10 enable
```

Disabling source learning on a VLAN causes the VLAN to be flooded with unknown unicast traffic.

Configuring VLAN IP Interfaces

Network device traffic is bridged (switched) at the Layer 2 level between ports that are assigned to the same VLAN. However, if a device needs to communicate with another device that belongs to a different VLAN, then Layer 3 routing is necessary to transmit traffic between the VLANs. Bridging makes the decision on where to forward packets based on the destination MAC address of the packet; routing makes the decision on where to forward packets based on the IP network address assigned to the packet (for example, 21.0.0.10).

The OmniSwitch supports the routing of IP traffic. A VLAN is available for routing when at least one IP interface is defined for that VLAN and at least one active port is associated with the VLAN. Up to eight IP interfaces can be configured for each VLAN.

If a VLAN does not have an IP interface, the ports associated with that VLAN are in essence firewalled from other VLANs. For information about configuring IP interfaces, see Chapter 17, “Configuring IP.”

Bridging VLANs Across Multiple Switches

To create a VLAN *bridging domain* that extends across multiple switches:

1 Create a VLAN on each switch with the same VLAN ID number (for example, VLAN 10).

2 On each switch, assign the ports that provide connections to other switches to the VLAN created in Step 1.

3 On each switch, assign the ports that provide connections to end user devices (for example, workstations) to the VLAN created in Step 1.

4 Connect switches and end user devices to the assigned ports.

The following diagram shows the physical configuration of an example VLAN bridging domain:

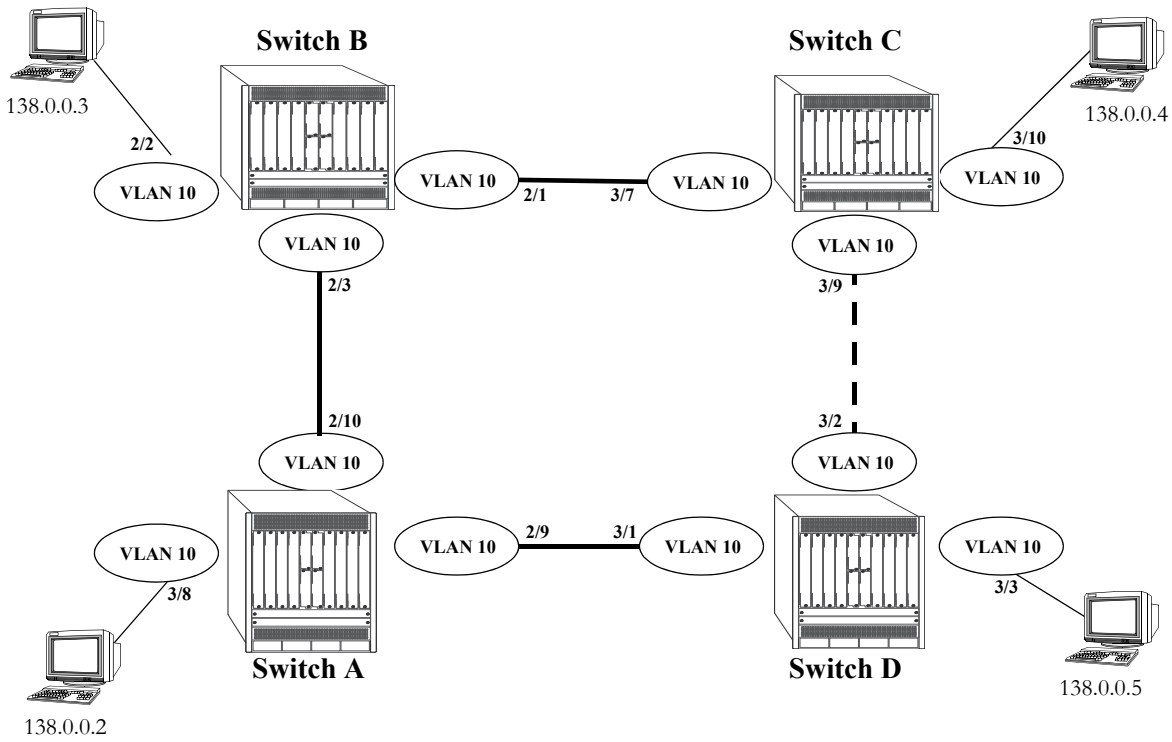


Figure 4-2 : VLAN Bridging Domain: Physical Configuration

In the above diagram, VLAN 10 exists on all four switches and the connection ports between these switches are assigned to VLAN 10. The workstations can communicate with each other because the ports to which they are connected are also assigned to VLAN 10. It is important to note that connection cables do not have to connect to the same port on each switch. The key is that the port must belong to the same VLAN on each switch. To carry multiple VLANs between switches across a single physical connection cable, use the 802.1Q tagging feature (see "Using 802.1Q Tagging" on page 4-7).

The connection between Switch C and D is shown with a broken line because the ports that provide this connection are in a blocking state. Spanning Tree is active by default on all switches, VLANs and ports. The Spanning Tree algorithm determined that if all connections between switches were active, a network loop would exist that could cause unnecessary broadcast traffic on the network. The path between Switch C and D was shut down to avoid such a loop. See Chapter 6, "Configuring Spanning Tree Parameters," for

information about how Spanning Tree configures network topologies that are loop free.

The following diagram shows the same bridging domain example as seen by the end user workstations. Because traffic between these workstations is *bridged* across physical switch connections within the VLAN 10 domain, the workstations are basically unaware that the switches even exist. Each workstation believes that the others are all part of the same VLAN, even though they are physically connected to different switches.

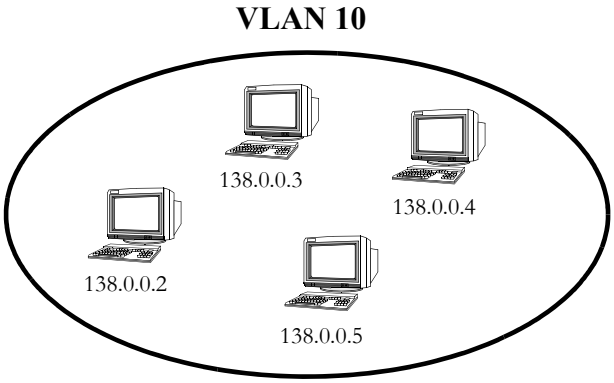


Figure 4-3 : VLAN Bridging Domain: Logical View

Creating a VLAN bridging domain across multiple switches allows VLAN members to communicate with each other, even if they are not connected to the same physical switch. This is how a logical grouping of users can traverse a physical network setup without routing and is one of the many benefits of using VLANs.

Verifying the VLAN Configuration

To display information about the VLAN configuration for a single switch use the show commands listed below:

show vlan	Displays a list of all VLANs configured on the switch and the status of related VLAN properties (for example, admin and Spanning Tree status and router port definitions).
show vlan members	Displays a list of VLAN port assignments.
show ip interface	Displays VLAN IP router interface information.

Understanding Port Output Display

Each line of the **show vlan members** output display corresponds to a single VLAN port association (VPA). In addition to showing the VLAN ID and slot/port number, the VPA type and current status of each association are also provided.

The VPA type indicates that one of the following methods was used to create the VPA:

Type	Description
default	The port was statically assigned to the VLAN using the **vlan members untagged** command. The VLAN is now the configured default VLAN for the port.
qtagged	The port was statically assigned to the VLAN using the **vlan members tagged** command. The VLAN is a static secondary VLAN for the 802.1Q tagged port.
mirror	The port is assigned to the VLAN because it is configured to mirror another port that is assigned to the same VLAN. For more information about the Port Mirroring feature, see Chapter 36, “Diagnosing Switch Problems.”

The VPA status indicates one of the following:

Status	Description
inactive	Port is not active (administratively disabled, down, or nothing connected to the port) for the VPA.
blocking	Port is active, but not forwarding traffic for the VPA.
forwarding	Port is forwarding all traffic for the VPA.
filtering	Mobile port traffic is filtered for the VPA; only traffic received on the port that matches VLAN rules is forwarded. Occurs when a mobile port’s VLAN is administratively disabled or the port’s default VLAN status is disabled. Does not apply to fixed ports.

The following example displays the VPA information for all ports in VLAN 200:

```
-> show vlan 200 members
  port     type      status
--------+---------+-------------
   3/24   default    inactive
   5/12   qtagged    blocking
```

The above example output provides the following information:

- VLAN 200 is the configured default VLAN for port 3/24, which is currently not active.
- VLAN 200 is an 802.1Q-tagged VLAN for port 5/12, which is an active port but currently blocked from forwarding traffic.

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Using Private VLANs

The Private VLAN (PVLAN) feature provides the ability to isolate Layer 2 data between devices that are on the same VLAN. This type of data isolation improves security and simplifies system configuration.

A standard VLAN usually represents a single broadcast domain, but a PVLAN divides a VLAN (Primary) into sub-VLANs (Secondary). The single broadcast domain is partitioned into smaller broadcast sub-domains while keeping the existing Layer 3 configuration. When a VLAN is configured as a PVLAN, the PVLAN is referred to as the Primary VLAN, and any subsequent VLANs that are associated with the Primary VLAN are referred to as Secondary VLANs.

For example, consider an example where a single switch is used by different work groups. The users from different work groups are all connected to the same VLAN. Having all the users operating in the same VLAN domain can lead to compromise in data security and complexity in managing them.

To isolate the users from each other, Secondary VLANs can be created for each work group under the Primary VLAN. The following diagram represents the scenario where W1, W2, and W3 are three different work groups sharing the same PVLAN. To isolate them from communicating with each other, they are assigned to individual Secondary VLANs. These Secondary VLANs cannot communicate with each other, except the PVLAN port. The PVLAN communicates with the promiscuous port and exchanges data with the respective Secondary VLANs.

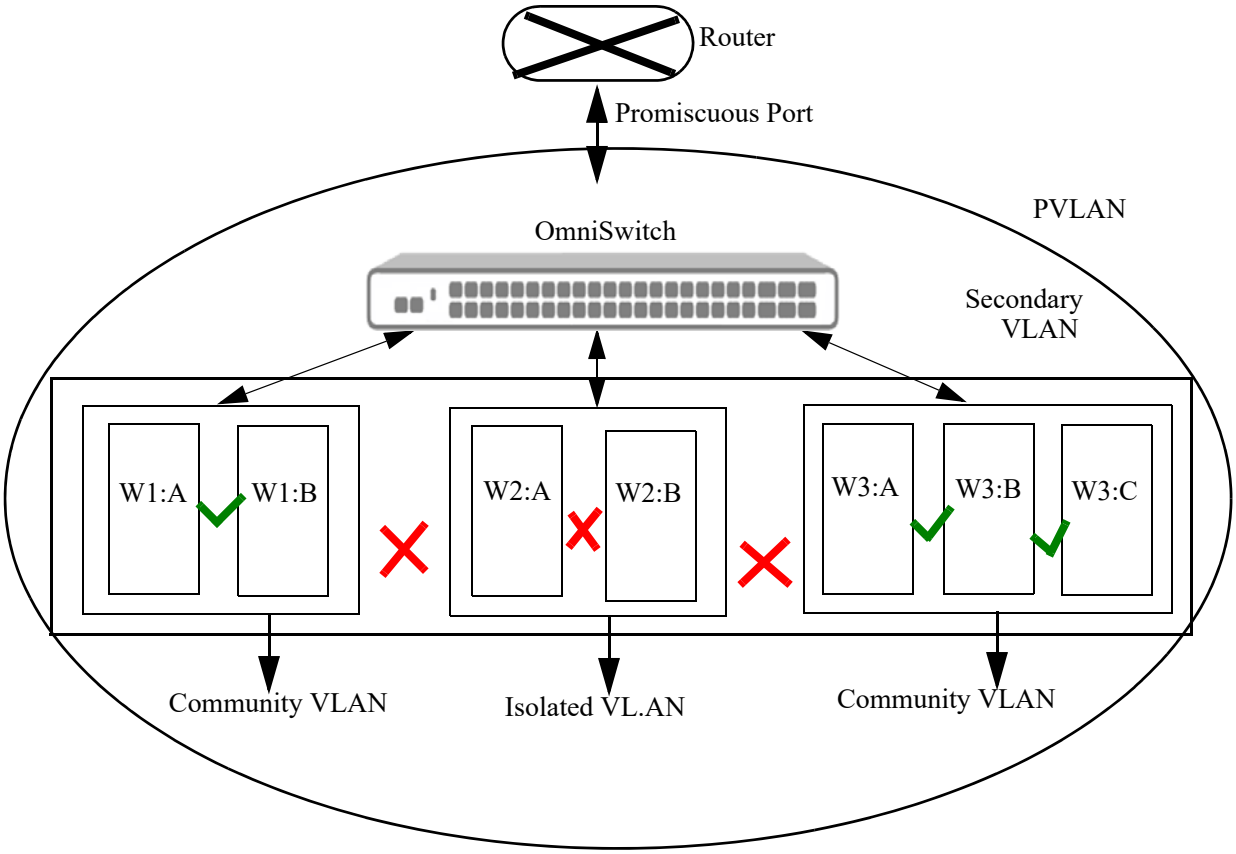


Figure 4-4 : Using Private VLANs

There are two types of Secondary VLANs:

- Isolated VLAN—In an Isolated VLAN, all hosts connected to a member port are Isolated at Layer 2. They can communicate only with the promiscuous port of the Primary VLAN. There can be only one Isolated VLAN within one Primary VLAN.
- Community VLAN—A Community VLAN is associated to a group of ports that connect to a certain "community" of end devices with mutual trust relationships. Any switch port associated with a common Community VLAN can communicate with each other and with the promiscuous ports of the Primary VLAN but not with any other Secondary VLAN. There can be multiple distinct Community VLANs within one Primary VLAN.

Private VLAN Ports

The ports with respect to PVLANs have different characteristics. The PVLAN port types are:

- PVLAN Isolated Port—An isolated port cannot communicate with any other port in the PVLAN except for promiscuous ports. This is a physical port or link aggregation port that is associated with an Isolated Secondary VLAN at the port level.
- PVLAN Community Port—A community port can only communicate with a promiscuous port and other ports that are part of the same Community VLAN in the same Primary VLAN. This is a physical port or link aggregation port that is associated with only one Community Secondary VLAN at the port level.
- PVLAN Promiscuous Port—The promiscuous port can communicate with all the isolated ports and community ports in the Primary VLAN. This is a physical port or link aggregation that is associated with only one Primary VLAN at the port level.
- PVLAN ISL Port—An inter-switch link port that extends a PVLAN domain across different switches by connecting Primary VLANs that belong to the same PVLAN domain. The ISL port carries both non-PVLAN traffic and Primary VLAN traffic between switches.

Quick Steps for Configuring PVLANs

The following steps provide a quick tutorial that creates a PVLAN. Also included are steps to define a Secondary VLAN for the PVLAN and assign ports to the PVLAN.

1 Create a PVLAN. Creating a PVLAN involves specifying a VLAN ID that is not already assigned to an existing VLAN. The specified VLAN ID will become the Primary VLAN for the PVLAN. For example, to create PVLAN 200 with the name "Corporate PVLAN" enter:

```
-> pvlan 200 name "Corporate PVLAN"
```

2 Enable the administrative state of PVLAN 200 by entering:

```
-> pvlan 200 admin-state enable
```

3 Create a Secondary VLAN and associate it to the Primary VLAN. Creating a Secondary VLAN involves specifying a VLAN ID that is not already assigned to an existing VLAN. The Secondary VLAN can be an Isolated VLAN or a Community VLAN depending on network requirements. For example, to create Isolated VLAN 250 and Community VLAN 251 and associate them to Primary VLAN 200, enter:

```
-> pvlan 200 secondary 250 type isolated
-> pvlan 200 secondary 251 type community
```

4 Associate the ports that will be part of the PVLAN. For example, to tag ports with Primary VLAN 200 and Secondary VLANs 250 and 251, enter:

```
-> pvlan 200 members port 1/1/1-3 tagged
-> pvlan 250 members port 1/1/10-12 tagged
-> pvlan 251 members port 1/1/20-22 tagged
```

Note. *Optional*. To verify the PVLAN configuration, use the **show pvlan** command. For example:

```
-> show pvlan
pvlan    type       admin   oper    mtu          name
------+----------+-------+------+------+-----------------
200    Primary    Ena     Dis    1500   PVLAN 200
250    Isolated   Ena     Dis    1500   PVLAN 250
251    Community  Ena     Dis    1500   PVLAN 251
```

To verify the mapping of Secondary VLANs to a Primary VLAN, use the **show pvlan mapping** command. For example:

```
-> show pvlan mapping
Primary    Secondary
VLAN       VLAN       Type
----------+----------+------------
200        250        Isolated
200        251        Community
```

To verify the port assignments for the PVLAN, use the **show pvlan members** command. For example:

```
-> show pvlan members
pvlan   port      type                status       port-type
-------+---------+------------------+------------+------------
200     1/1       qtagged             inactive     promiscuous
200     1/2       qtagged             inactive     promiscuous
200     1/3       qtagged             inactive     promiscuous
250     1/10      qtagged             inactive     isolated
250     1/11      qtagged             inactive     isolated
250     1/12      qtagged             inactive     isolated
251     1/20      qtagged             inactive     community
251     1/21      qtagged             inactive     community
251     1/22      qtagged             inactive     community
```

PVLAN Management Overview

The PVLAN feature provides the ability to create Secondary VLANs within a Primary VLAN. A regular VLAN usually represents a single broadcast domain. However, a PVLAN divides a VLAN (Primary) into sub-VLANs (Secondary) to partition the single broadcast domain into smaller broadcast sub-domains while keeping the existing Layer 3 configuration.

The ports can be isolated from each other at the data link layer to improve security and performance and also simplify IP address assignment. The following PVLAN configuration tasks can be performed on the switch:

- Create a Primary VLAN, see page 4-18.
- Create Secondary VLANs (Community or Isolated) to associate with a Primary VLAN, see page 4-19.
- Configure a user port or a link aggregate as a promiscuous port or ISL port for a Primary VLAN, see page 4-20.

- Associate the Secondary VLANs to user ports or link aggregates, see page 4-21.
- Verify the PVLAN configuration, see page 4-25.

Creating PVLANs

Before creating a PVLAN, consider the following points:

- A Primary VLAN ID is created first and represents the PVLAN domain. When any Secondary VLANs are created, the Primary VLAN ID must be specified to identify the PVLAN to which the Secondary VLAN is assigned.
- The VLAN ID that is specified to create a Primary VLAN must *not* already exist in the system.
- The following values are configurable only on the Primary VLAN but are also applied to all the Secondary VLANs that are associated with the Primary VLAN:
  - The administrative status for the PVLAN (enabled by default). When the status is changed for the Primary VLAN ID, the change is automatically applied to the Secondary VLANs.
  - The Spanning Tree status for the PVLAN (enabled by default). When the status is changed for the Primary VLAN ID, the change is automatically applied to the Secondary VLANs.
  - IP configuration for the PVLAN. An IP interface is configured on the Primary VLAN but is not configurable on Secondary VLANs.
- Specifying a description for Primary and Secondary VLANs is optional. If one is not specified, then the VLAN ID is used as the description.
- MVRP cannot be enabled on the PVLAN.

To create a Primary VLAN for the PVLAN, enter **pvlan** followed by a unique VLAN ID, an optional administrative status, and an optional description. For example, the following command creates Primary VLAN 200 with a description:

```
-> pvlan 200 admin-state enable name "Corporate PVLAN"
```

When configuring a description that contains multiple words that are separated by spaces, quotations marks are required. If the description consists of only one word or multiple words separated by another character (such as a hyphen), then quotation marks are not required.

You can also specify a range of VLAN IDs with the **pvlan** command. Use a hyphen to indicate a contiguous range. For example, the following commands create Primary VLANs 10 through 15 and 100 through 105 on the switch:

```
-> pvlan 10-15 100-105 name "Corporate PVLAN"
-> pvlan 100-105 name "Corporate PVLAN"
```

The maximum transfer unit (MTU) size can also be configured for the PVLAN when the PVLAN is created. For example, to set an MTU size of 64 KB for PVLAN 200, enter the following:

```
-> pvlan 200 mtu-ip 64
```

To remove a PVLAN from the switch configuration, use the **no** form of the **pvlan** command. For example, to remove PVLANs 10-15, 100-105, and 200, enter:

```
-> no pvlan 10-15
-> no pvlan 100-105
-> no pvlan 200
```

When the Primary VLAN for a PVLAN is deleted, any router interfaces defined for the PVLAN are removed and all VLAN port associations are dropped.

To view a list of PVLANs already configured on the switch, use the **show pvlan** command. See "Verifying the PVLAN Configuration" on page 4-25 for more information.

Enabling/Disabling the PVLAN Administrative Status

The administrative state of a PVLAN is enabled by default. To enable or disable the administrative status for an existing PVLAN, enter **pvlan** followed by an existing Primary VLAN ID and either **admin-state enable** or **admin-state disable**. For example:

```
-> pvlan 200 admin-state enable
-> pvlan 200 admin-state disable
```

When the administrative status for the Primary VLAN of a PVLAN is changed, the following occurs:

- The change is automatically made to any Secondary VLANs associated with the Primary VLAN.
- PVLAN port assignments are retained but traffic is not forwarded on these ports if the administrative status is disabled.

Modifying the PVLAN Description

To change the description for the Primary VLAN of a PVLAN, enter **pvlan** followed by an existing VLAN ID and the keyword **name** followed by the new description (up to 32 characters). For example, the following command changes the description for Primary VLAN 200 to "Corporate IP Network":

```
-> pvlan 455 name "Corporate IP Network"
```

Creating Secondary VLANs

Before creating Secondary VLANs for a PVLAN, consider the following points:

- The VLAN ID used to configure the Secondary VLAN must *not* already exist in the system.
- The Secondary VLAN can be created only after the Primary VLAN for the PVLAN is created.
- There are two types of Secondary VLANs: Isolated and Community. Only one Isolated VLAN can be associated with a Primary VLAN, but multiple Community VLANs can be associated with the same Primary VLAN.
- The administrative state of Secondary VLANs is derived from the administrative state of the Primary VLAN.
- The Spanning Tree state of Secondary VLANs is derived from the Spanning Tree state of the associated Primary VLAN.
- MVRP cannot be enabled on a Secondary VLAN.

To create and associate a Secondary VLAN to a Primary VLAN, use the **pvlan secondary** command. For example, the following commands create Isolated and Community VLANs for Primary VLAN 200:

```
-> pvlan 200 secondary 250 type isolated
-> pvlan 200 secondary 251 type community
```

By default, the administrative status and the Spanning Tree status of the associated Primary VLAN is applied to both of the configured Secondary VLANs.

You can also specify a range of Secondary VLAN IDs when creating Community VLANs. Use a hyphen to indicate a contiguous range and a space to separate multiple VLAN ID entries. For example, the following command creates and associates Community VLANs 20 through 25 to Primary VLAN 200 on the switch:

-> pvlan 200 secondary 20-25 type community

Specifying a range of Secondary VLAN IDs is not allowed when creating Isolated VLANs. There is only one Isolated VLAN allowed for each Primary PVLAN.

To remove a Secondary VLAN from the PVLAN, use the **no** form of the **pvlan secondary** command. For example:

```
-> no pvlan 200 secondary 251
```

When a Secondary VLAN is deleted, the VLAN ID is removed and all VLAN port associations for that VLAN are dropped.

To view the PVLAN mapping of Secondary VLANs configured on the switch, use the **show pvlan mapping** command. See "Verifying the PVLAN Configuration" on page 4-25 for more information.

Assigning Ports to PVLANs

PVLAN offers Layer 2 data isolation between the devices on the same VLAN. For a PVLAN to operate, ports or link aggregates must be assigned to the PVLAN. The following port types are configurable for PVLANs:

- Promiscuous Port
- ISL Port
- Isolated Port
- Community Port

An ISL port can only be assigned to a Primary VLAN. The other port types are determined based on the type of VLAN associated with the PVLAN to which the port is assigned. For example:

- Ports assigned to a Primary PVLAN are designated as promiscuous ports.
- Ports assigned to a Secondary VLAN configured as a Community VLAN are designated as community ports.
- Ports assigned to a Secondary VLAN configured as an Isolated VLAN are designated as isolated ports.

The following table defines the communication criteria between the PVLAN port types:

Port Types	**Isolated**	**Promiscuous**	**Community**	**ISL**
Isolated	Deny	Permit	Deny	Permit
Promiscuous	Permit	Permit	Permit	Permit
Community	Deny	Permit	Deny, except for ports that belong to the same Community.	Permit

ISL	Only PVLAN packets	Permit	Only PVLAN packets and the same Community VLAN packets	Permit

Configuring Promiscuous Ports

A PVLAN must have one promiscuous port associated with the Primary VLAN to communicate with all the community ports, isolated ports, and ISL ports.

A promiscuous port can be tagged or untagged based on the network requirements.

To configure a promiscuous port, use the **pvlan members** command to assign a port or link aggregate as a tagged or untagged member of the Primary VLAN. For example:

```
-> pvlan 200 members port 1/1/1 tagged
```

In this example, the port 1/1/1 is assigned to Primary PVLAN 200, so the port is designated as a promiscuous port. When only a tagged VLAN-port association (VPA) is configured, then all untagged traffic is dropped on the port.

To remove a promiscuous port from the Primary VLAN, use the **no** form of the **pvlan members** command. For example:

```
-> no pvlan 200 members port 1/1/1
```

Configuring ISL Ports

An Inter-Switch-Link (ISL) port connects a Primary VLAN on one switch to a Primary VLAN on another switch to extend the PVLAN domain across multiple switches. The ISL port carries Primary and Secondary VLAN traffic between switches throughout the PVLAN domain. Make sure that the Primary and Secondary VLAN configuration is the same across all the switches to ensure the traffic is forwarded correctly over the ISL connections.

To configure an ISL port, use the **pvlan members** command with the **isl** parameter option. For example, the following command configures port 1/1/2 as an ISL port for Primary VLAN 200:

```
-> pvlan 200 members port 1/1/2 isl
```

Note. An ISL port can be configured only on the Primary VLAN, but the ISL port carries traffic for all VLANs associated with the PVLAN (Primary and Secondary).

To remove an ISL port from the Primary VLAN, use the **no** form of the **pvlan members** command. For example:

```
-> no pvlan 200 members port 1/1/2
```

Configuring Secondary VLAN Ports

The Secondary VLAN ports are defined as isolated or community ports based on the type of Secondary VLAN ID to which the ports are assigned.

- If a port is assigned to an Isolated Secondary VLAN, the port is designated as an isolated port.
- If a port is assigned to a Community Secondary VLAN, the port is designated as a community port.

To configure an isolated port, use the **pvlan members** command to assign a port or link aggregate as a tagged or untagged member of an Isolated Secondary VLAN. For example, the following commands create Isolated VLAN 250 as a Secondary VLAN to Primary VLAN 200 and then assign port 1/2/2 and link aggregate 10 to VLAN 250:

```
-> pvlan 200 secondary 250 type isolated
-> pvlan 250 members port 1/2/2 tagged
-> pvlan 250 members linkagg 10 untagged
```

To configure a community port, use the **pvlan members** command to assign a port or link aggregate as a tagged or untagged member of a Community Secondary VLAN. For example, the following commands create Community VLAN 251 as a Secondary VLAN to Primary VLAN 200 and then assign port 1/2/5 and link aggregate 15 to VLAN 251:

```
-> pvlan 200 secondary 251 type isolated
-> pvlan 251 members port 1/2/5 tagged
-> pvlan 251 members linkagg 15 untagged
```

To remove a port from a Secondary VLAN, use the **no** form of the **pvlan members** command. For example, the following commands remove a port and link aggregate from Secondary VLAN 250 and 251:

```
-> no pvlan 250 members port 1/2/2
-> no pvlan 251 members linkagg 15
```

Assigning UNP Ports to Secondary VLANs

Universal Network Profile (UNP) ports can also be assigned to Secondary VLANs (isolated or community ports). The UNP ports are designated as isolated or community ports during runtime based on the first MAC address learned on the port.

- If the first MAC address is learned on a UNP port is classified into an Isolated VLAN, the port is designated as an isolated port.
- If the first MAC address is learned on a UNP port is classified into a Community VLAN, the port is designated as a community port.
- If the first MAC address learned on the a UNP port is classified into any standard VLAN (non-PVLAN), then the UNP port cannot be designated as an isolated or community port.

Protocol Configuration Requirements for PVLAN

This section contains important information about configuring other protocols to interact with PVLANs. For more information about each protocol, refer the related chapters in the *OmniSwitch AOS Release 8 CLI Reference Guide* and the *OmniSwitch AOS Release 8 Network Configuration Guide*.

Enabling DHCP Snooping for PVLANs

DHCP Snooping can be enabled only on the Primary VLAN of a PVLAN configuration. When enabled on the Primary VLAN, the configuration will be applied to the Secondary VLANs associated with the Primary VLAN.

If the DHCP Snooping server is on another chassis, then the ISL port configured for communication must be configured as a trusted port.

Enabling Ingress Source Filtering (ISF) for PVLANs

ISF can be enabled only on the Primary VLAN of a PVLAN configuration. When enabled on the Primary VLAN, the configuration will be applied to the Secondary VLANs associated with the Primary VLAN.

Enabling IPMS for PVLANs

IPMS can be enabled only on the Primary VLAN of a PVLAN configuration.

Enabling STP for PVLANs

STP can be enabled only on the Primary VLAN of a PVLAN configuration. When enabled on the Primary VLAN, the configuration will be applied to the Secondary VLANs associated to the Primary VLAN.

Note. The PVLAN feature is supported only when the switch is running in the flat (MSTP) Spanning Tree mode; it is not supported in the per-VLAN mode.

Sample PVLAN Use Case

PVLAN Spanning across Multiple Systems

The following diagram shows how using a PVLAN configuration allows the traffic to be segmented at the Layer 2 level, thus limiting the broadcast domain and extending it across multiple switches.

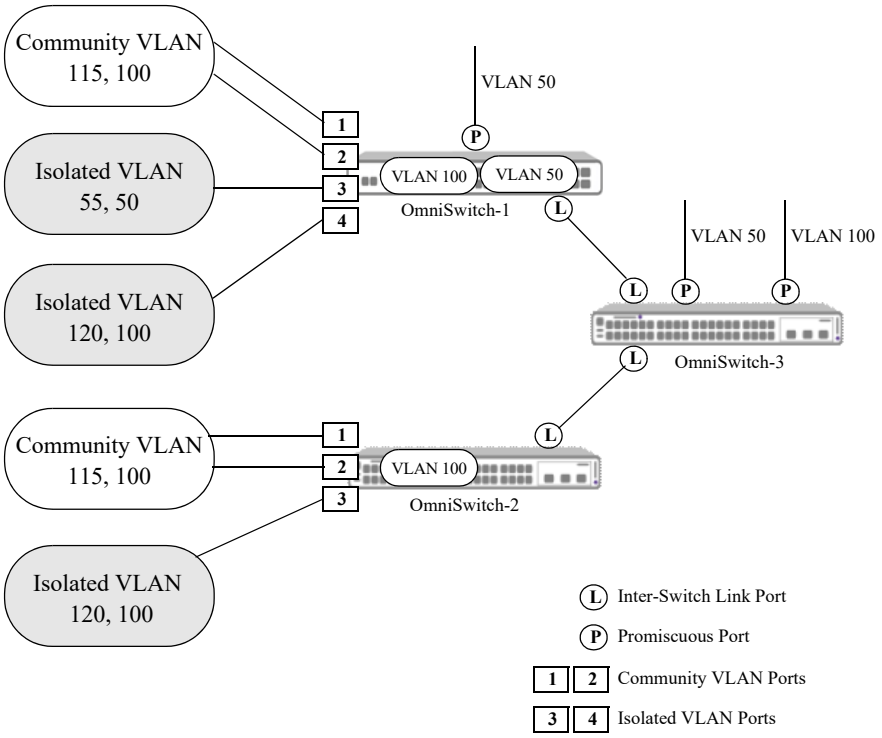


Figure 4-5 : PVLAN Spanning across Multiple Systems

The individual switches are separately configured with the PVLAN setup. IP interfaces are configured on the Primary VLAN, and the hosts in both the Isolated and Community VLAN can share the IP addresses from the same subnet but still remain isolated.

In this use case example, there are two Primary VLANs (100, 50) spanning across multiple OmniSwitch systems:

Primary VLAN: 100 (IP subnet: 10.10.100.x)

- Community VLAN: 115
- Isolated VLAN: 120

Primary VLAN: 50 (IP subnet: 10.10.50.x)

- Isolated VLAN: 55

All the isolated, community, and promiscuous ports can be untagged or tagged. Since the PVLAN domain spans across multiple switches, an Inter-Switch Link (ISL) port is configured for each Primary VLAN on each switch to connect and carry traffic forwarded on the Primary VLANs.

The PVLAN traffic flow in this scenario is as follows:

- Untagged traffic is passed into an untagged Secondary (community) port 1 on OmniSwitch-1.
  - The traffic will be tagged with the PVID of the port which is Secondary VLAN.
  - The ISL port will then carry the tagged traffic into the community port on the other switch (OmniSwitch-2:1, OmniSwitch-2:2).
  - Traffic outgoing through the promiscuous port (OmniSwitch-3: P: 100) will modify the tag to Primary VLAN.
- Tagged traffic is passed into an untagged Secondary (community) port
  - If the tag matches the PVID of the port, it will be allowed.
  - The ISL port will then carry the tagged traffic into the community port on the other switch (OmniSwitch-2:1, OmniSwitch-2:2).
  - Traffic outgoing through promiscuous port (OmniSwitch-3: P: 100) will modify the tag to Primary VLAN.
- Untagged traffic passed into a tagged Secondary (community) port is dropped.
- Tagged traffic passed into a tagged Secondary (community) port is dropped if the VLAN tag of the traffic does not match the VLAN tag of the port.

Verifying the PVLAN Configuration

To display information about the PVLAN configuration, use the **show** commands listed below:

show pvlan	Displays a list of PVLANs configured on the switch.
show pvlan mapping	Displays Primary PVLAN and Secondary PVLAN mapping.
show pvlan members	Displays port associations (VPAs) for all or specific PVLANs.

Use the **show configuration snapshot** command with the **pvlan** option to display the PVLAN configuration. For example:

```
-> show configuration snapshot pvlan
! PVLAN:
pvlan 300 admin-state enable
pvlan 300 secondary 20-25 type community
pvlan 300 members port 1/3/10 tagged
pvlan 300 members linkagg 10 tagged
```

5 Configuring High Availability VLANs

High availability (HA) VLANs, unlike standard VLANs, allow you to send traffic intended for a single destination MAC address to multiple switch ports. These high availability VLANs can be used to manage server clusters.

In This Chapter

This chapter describes the basic components of high availability VLANs and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Creating a VLAN on page 5-6.
- Adding and Removing Server Cluster Ports to a HA VLAN on page 5-7.
- Assigning and Modifying Server Cluster Mode on page 5-7.
- Assigning and Removing MAC addresses to a HA VLAN on page 5-8.

High Availability Default Values

The table below lists default values for high availability VLAN software.

Parameter Description	Command	Default Value/Comments
Server cluster admin state of the server cluster	**server-cluster**	admin-state - enable
Server cluster id and mode	**server-cluster**	mode - L2
Mac address of the server cluster	**server-cluster mac-address**	None
IP address of the server cluster	**server-cluster ip**	IP address is configurable only for L3 clusters.
Configure the port/linkagg of a server cluster	**server-cluster port** **server-cluster linkagg**	None

Quick Steps for Creating High Availability VLANs

Follow the steps below for a quick tutorial on configuring high availability (HA) VLANs. Additional information on how to configure each command is given in the sections that follow.

1 Create a server cluster that will become the HA VLAN by using the command **server-cluster** and configure the mode. For example:

```
-> server-cluster 1 name l2_cluster mode l2
```

2 Create a default VLAN for the HA VLAN ports with the **vlan** command as shown below:

```
-> vlan 10
```

3 Assign member ports to the new default VLAN with the **vlan members untagged** command as shown below:

```
-> vlan 10 members port 1/3 untagged
-> vlan 10 members port 1/4 untagged
-> vlan 10 members port 1/5 untagged
```

4 Assign mac-address for the new server cluster by using the command **server-cluster mac-address**. For example:

```
-> server-cluster 1 vlan 10 port 1/3-5 mac-address 01:00:11:22:33:44
```

Note. *Optional*. You can display the configuration of high availability VLANs with the **show server-cluster** command. For example:

```
-> show server-cluster 1
Cluster Id : 1,
Cluster Name : L2-cluster,
Cluster Mode : L2,
Cluster Mac-address : 01:10:11:22:33:44,
Cluster Vlan : 12,
Administrative State: Enabled,
Operational State : Disabled,
Operational Flag : VPA is not forwarding
```

An example of what these commands look like entered sequentially on the command line:

```
-> server-cluster 1 mode L2
-> vlan 10
-> vlan 10 members port 1/3 untagged
-> vlan 10 members port 1/4 untagged
-> vlan 10 members port 1/5 untagged
-> server-cluster 1 vlan 10 port 1/3-5 mac-address 01:00:11:22:33:44
```

High Availability VLAN Overview

High availability (HA) VLANs send traffic intended for a single destination MAC address to multiple switch ports. An HA VLAN is configured by creating a standard VLAN and then assigning ports to the VLAN. Once these types of ports are assigned, the standard VLAN automatically becomes an HA VLAN. When this occurs, standard VLAN commands no longer apply.

Destination MAC addresses (unicast and multicast) are also assigned to high availability VLANs. These addresses identify ingress port traffic that the switch will send out on all egress ports that belong to the same VLAN

In addition to assigning ingress and egress ports, tagging inter-switch link ports with an HA VLAN ID is allowed. Ingress port traffic destined for an HA VLAN MAC address is sent out on all egress *and* inter-switch link ports that belong to the same VLAN. Traffic forwarded on inter-switch link ports is done so in accordance with the Spanning Tree state of the port.

A high availability VLAN hosts multiple instances of applications like e-commerce applications, critical databases, business applications etc and supports redundancy. Each instance may get all the service requests and based on a shared algorithm, HA VLAN decides on which requests a particular node has to handle. Apart from service request paths, the nodes are internally connected to share information related to the service load information, service request data and service availability on other nodes.

The HA VLAN feature on the OmniSwitch provides an elegant and flexible way to connect the server cluster nodes directly to the ingress network. This involves multicasting the service requests on the configured ports. The multicast criteria is configurable based on destination MAC and destination IP address. Egress ports can be statically configured on a server cluster or they can be registered by IGMP reports. The server cluster feature on the OmniSwitch multicast the incoming packets based on the server cluster configuration on the ports associated with the server cluster.

High Availability VLAN Operational Mode

There are typically two modes of implementation of server clusters in HA VLAN.

- Layer 2 - The server cluster is attached to a L2 switch on which the frames destined to the cluster MAC address are to be flooded on all interfaces. For more information see "Example 1: Layer 2 Server Cluster" on page 5-9
- Layer 3 - The server cluster is attached to a L3 switch on which the frames destined to the server cluster IP address are to be routed to the server cluster IP and then flooded on all interfaces. For more information see "Example 2: Layer 3 Server Cluster" on page 5-11.

Note. The L2 mode is currently supported in AOS using the static mac-address command and L3 mode by the static ARP command.

Traffic Flows in High Availability VLAN

The figure below shows how ingress traffic is handled by high availability VLANs.

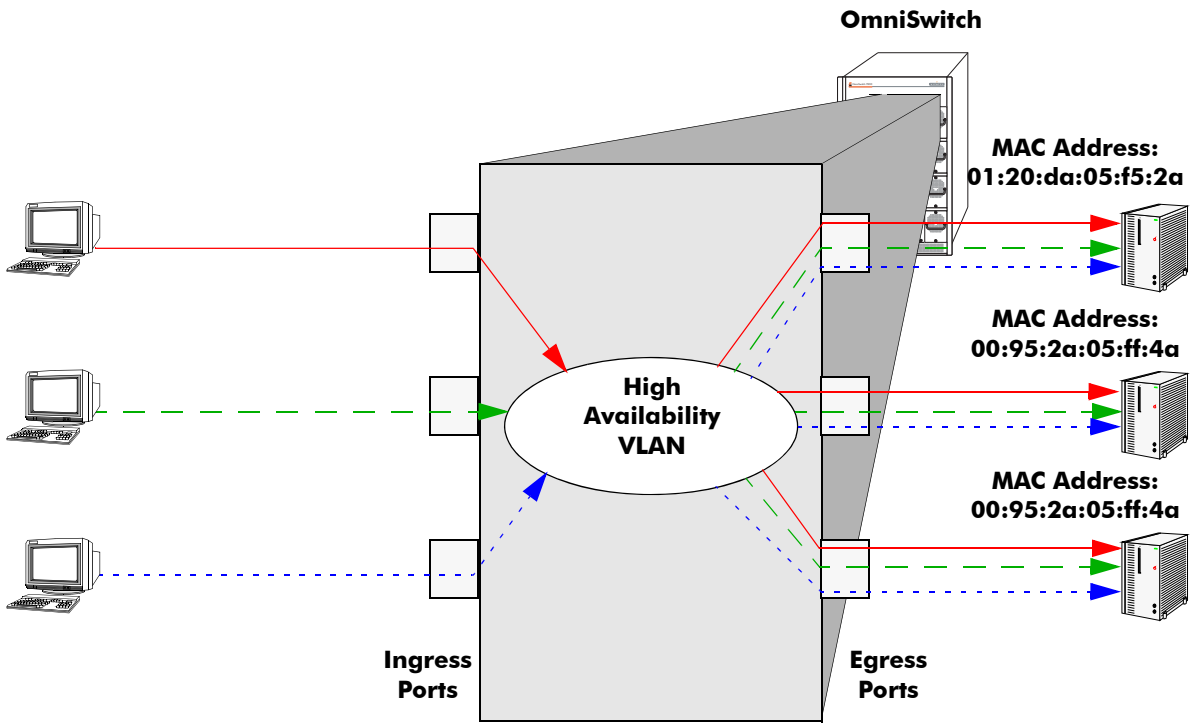


Figure 5-1 : Example of an L2 Server Cluster - Ingress to Egress Port Flow

In the above example, packets received on the ingress ports that are destined for the high availability VLAN MAC address are sent out the egress ports that are members of the same VLAN. The MAC address is virtual to the server cluster, individual servers may have different physical MAC address.Since all three servers are connected to egress ports, they all receive the ingress port traffic. This provides a high level of availability in that if one of the server connections goes down, the other connections still forward traffic to one of the redundant servers.

Configuring High Availability VLANs on a Switch

This section describes how to use the Command Line Interface (CLI) commands to configure high availability (HA) VLANs on a switch. For a brief tutorial on configuring HA VLANs, see "Quick Steps for Creating High Availability VLANs" on page 5-3.

When configuring HA VLANs, you must perform the following steps:

1 Create a VLAN. To create a VLAN use the **vlan** command, which is described in "Creating and Deleting VLANs" on page 5-6.

2 Assign VLAN member ports. To assign member ports to the VLAN, use the **vlan members untagged** command which is described in "Changing the Default VLAN Assignment for a Port" on page 4-6.

3 Create a server cluster and configure the mode. To create a server cluster and configure the cluster mode, use the **server-cluster** command which is described in "Adding and Removing Server Cluster Ports" on page 5-7.

4 Assign MAC Addresses. To assign MAC addresses to the HA VLAN server cluster, use the **server-cluster mac-address** command, which is described in "Assigning and Removing MAC Addresses" on page 5-8.

Note. Use the **show server-cluster** command to verify the HA VLAN configuration on the switch. See "Displaying High Availability VLAN Status" on page 5-16 for more information.

Creating and Deleting VLANs

The following subsections describe how to create and delete a VLAN with the **vlan** command.

Note. This section provides only a basic description of creating and deleting VLANs. For a complete description of configuring and monitoring VLANs on a switch, please refer to Chapter 4, "Configuring VLANs."

Creating a VLAN

To create a new VLAN use the **vlan** command by entering **vlan** followed by the VLAN ID number. For example, to create a VLAN with a VLAN ID number of 10 enter:

```
-> vlan 10
```

You can also specify the administrative status and a name for the VLAN with the **vlan** command. For example, to administratively enable (the default) a VLAN when you configure it enter **vlan** followed by the VLAN ID number and **enable**.

For example, to create VLAN 10 and administratively enable it enter:

```
-> vlan 10 enable
```

Deleting a VLAN

To delete a VLAN use the **no** form of the **vlan** command by entering **no vlan** followed by the VLAN's ID number. For example, to delete high availability VLAN 10 enter:

```
-> no vlan 10
```

Adding and Removing Server Cluster Ports

The following subsections describe how to assign to and remove ingress ports from a high availability VLAN with the **server-cluster port** command.

Assigning Ports to a Server Cluster

To assign server cluster ports to a high availability VLAN use the **server-cluster port/linkagg** command. For example, to assign port 1/21 to server cluster "1", enter the commands as:

```
-> server-cluster 1 port 1/21
```

To assign linkagg "1" to server cluster "3', enter the commands as:

```
-> server-cluster 3 linkagg 1
```

Removing Ports from a Server Cluster

To remove server cluster ports from a high availability VLAN use the **no** form of **server-cluster port/ linkagg** command. For example,

```
-> no server-cluster 1 port 1/21
-> no server-cluster 3 linkagg 1
```

Assigning and Modifying Server Cluster Mode

The following subsections describe how to assign to and remove egress ports from a high availability VLAN with the **server-cluster** command.

Assigning L2 Mode to a Server Cluster

To assign L2 mode to a high availability VLAN use the **server-cluster id** command. For example, to assign "L2" mode to the server cluster "1", enter the command as:

```
-> server-cluster 1 mode l2
```

If you want a name to be assigned along with the cluster mode, enter the commands as:

```
-> server-cluster 1 name l2_cluster mode l2
```

Assigning L3 Mode to a Server Cluster

A cluster can be assigned an IP address and an ARP entry mac-address. Each cluster should have a unique IP-address. IP address is configurable only for L3 clusters.

To assign L3 mode to a high availability VLAN use the **server-cluster id** command. For example, to assign "L3" mode to the server cluster "2", enter the command as:

```
-> server-cluster 2 mode l3
-> server-cluster 5 port all
```

To assign L3 mode to linkaggs, enter the commands as:

```
-> server-cluster 3 linkagg 1
-> server-cluster 4 linkagg 1-3
```

To remove server cluster from a high availability VLAN, use the **no** form of the command. For example,

```
-> no server-cluster 1
-> no server-cluster 2
```

Assigning and Removing MAC Addresses

The following subsections describe how to assign and remove MAC addresses from a high availability VLAN with the **server-cluster mac-address** command. Traffic that is received on ingress ports that contains a destination MAC address that matches the high availability VLAN address is sent out all egress ports that belong to the high availability VLAN.

Note. The multicast addresses within the following reserved ranges are not supported:
- 01:00:5E:00:00:00 to 01:00:5E:7F:FF:FF
- 01:80:C2:XX.XX.XX
- 33:33:XX:XX:XX:XX.

Assigning MAC Addresses

To assign a MAC address to a high availability VLAN, use the **server-cluster mac-address** command by entering **server-cluster mac-address**, followed by the VLAN's ID number, **mac**, and the MAC address. Note that both unicast and multicast addresses are supported.

For example, to assign the MAC address 00:25:9a:5c:2f:10 to high availability VLAN 20, enter the command as:

```
-> server-cluster mac-address vlan 20 mac 00:25:9a:5c:2f:10
```

To add more than one MAC address to a high availability VLAN, enter each address on the same command line separated by a space. For example, to assign MAC addresses 00:25:9a:5c:2f:11, 00:25:9a:5c:12, and 01:00:00:3f:4c:10, to high availability VLAN 30, enter the command as:

```
-> server-cluster mac-address vlan 30 mac 00:25:9a:5c:2f:11 00:25:9a:5c:12
01:00:00:3f:4c:10.
```

Removing MAC Addresses

To remove a MAC address associated with a high availability VLAN, use the **no** form of the **server-cluster mac-address** command. For example, the following command removes MAC address 00:25:9a:5c:2f:10 from VLAN 20:

```
-> no server-cluster mac-address vlan 20 no mac 00:25:9a:5c:2f:10
```

To remove more than one MAC address from a high availability VLAN using a single command, enter each address on the same command line separated by a space. For example, to remove MAC addresses 00:25:9a:5c:2f:11, 00:25:9a:5c:12, and 01:00:00:3f:4c:10, from high availability VLAN 30, enter the command as:

```
-> server-cluster mac-address vlan 30 no mac 00:25:9a:5c:2f:11 00:25:9a:5c:12
01:00:00:3f:4c:10.
```

Note. Removing the last MAC address from an HA VLAN is not allowed. Deleting the VLAN is required when there is only one MAC address left.

Application Examples

This section contains the following HAVLAN application examples:

- "Example 1: Layer 2 Server Cluster" on page 5-9.
- "Example 2: Layer 3 Server Cluster" on page 5-11.
- "Example 3: Layer 3 Server Cluster with IP Multicast Address to Cluster (IGMP)" on page 5-13.

Example 1: Layer 2 Server Cluster

In the following example, the MAC address can be unicast or L2 multicast or IP multicast.

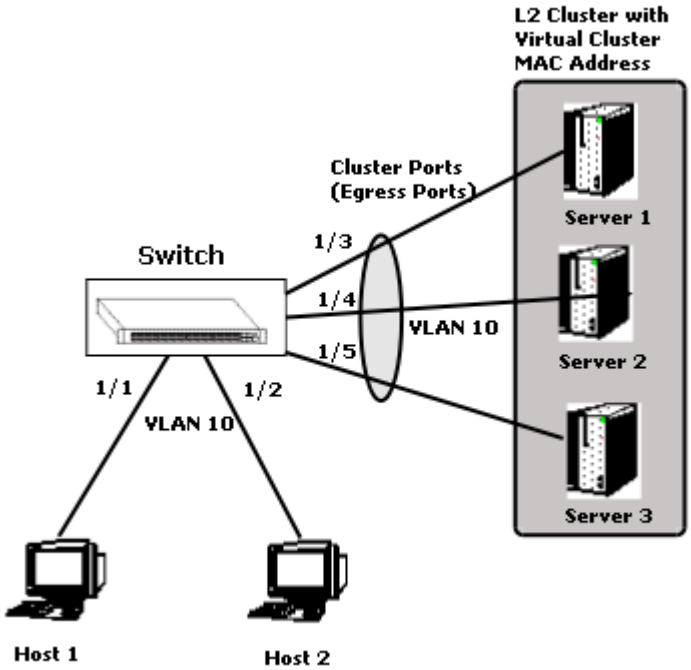


Figure 5-2 : Switch connected to an L2 Server Cluster through 3 ports (1/3, 1/4, 1/5)

- A server cluster can be configured with a unique MAC address and a VLAN with a port list
- The traffic which ingresses on 1/1 or 1/2 destined to the server cluster MAC address and the VLAN is forwarded to all the egress ports configured.(1/3,1/4,1/5).
- Here the ingress ports must be in the same VLAN as the server cluster VLAN and egress ports and other traffic must be switched according to the normal switching logic.

Configuration Example

In this example, a packet can be an L2 or IP switched packet and Egress port can also be a linkagg port.

1 Create a server cluster that will become the HA VLAN by using the command **server-cluster** and configure the mode. For example:

```
-> server-cluster 1 mode l2 admin-state enable
```

2 Create a default VLAN for the HA VLAN ports with the **vlan** command as shown below:

```
-> vlan 10
```

3 Assign member ports to the new default VLAN with the **vlan members untagged** and **server-cluster** commands as shown below:

```
-> vlan 10 members port 1/3 untagged
-> vlan 10 members port 1/4 untagged
-> vlan 10 members port 1/5 untagged
-> server-cluster 1 port 1/3
-> server-cluster 1 port 1/4
-> server-cluster 1 port 1/5
```

4 Assign mac-address for the new server cluster by using the command **server-cluster mac-address**. For example:

```
-> server-cluster 1 vlan 10 port mac-address 01:00:11:22:33:44
```

Note. *Optional*. You can display the configuration of high availability VLANs with the **show server-cluster** command. For example:

```
-> show server-cluster 1
Cluster Id : 1,
Cluster Name : L2-cluster,
Cluster Mode : L2,
Cluster Mac-address : 01:10:11:22:33:44,
Cluster Vlan : 12,
Administrative State: Enabled,
Operational State : Disabled,
Operational Flag : VPA is not forwarding
```

An example of what these commands look like entered sequentially on the command line:

```
-> server-cluster 1 mode L2 admin-state enable
-> vlan 10
-> vlan 10 members port 1/3 untagged
-> vlan 10 members port 1/4 untagged
-> vlan 10 members port 1/5 untagged
-> server-cluster 1 port 1/3
-> server-cluster 1 port 1/4
-> server-cluster 1 port 1/5
-> server-cluster 1 vlan 10 port 1/3-5 mac-address 01:00:11:22:33:44
```

Example 2: Layer 3 Server Cluster

In this example, A server cluster is configured with a unique IP address and a static ARP entry (cluster MAC) and a port list. Here, the server cluster IP address must be a unicast address.

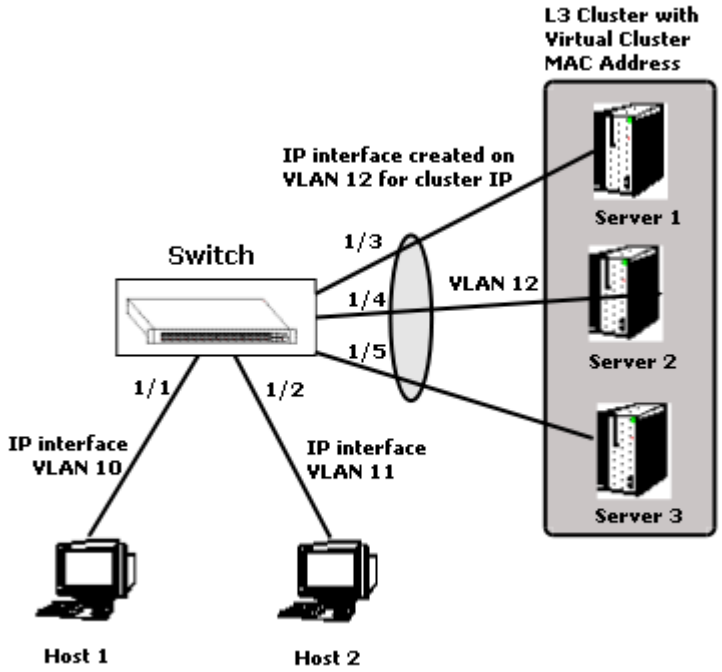


Figure 5-3 : Switch connected to an L3 Server Cluster through 3 ports (1/3,1/4,1/5)

- The traffic which ingresses on 1/1 or 1/2 destined to the server cluster IP is routed to all the egress ports configured (1/3,1/4,1/5). The ingress ports are on a different VLAN as the server cluster IP interface.
- However, all the egress ports need to be in the same VLAN as the IP interface of server cluster. The other traffic must be switched according to the normal switching/routing logic.
- Egress port can be a linkagg port as well.

Configuration Example

In this example, a packet is an L3 or IP switched packet.

1 Create a server cluster that will become the HA VLAN by using the command **server-cluster** and configure the mode. For example:

```
-> server-cluster 2 mode L3 admin-state enable
```

2 Create a default VLAN for the HA VLAN ports with the **vlan** command as shown below:

```
-> vlan 12
```

3 Assign member ports to the new default VLAN with the **vlan members untagged** and **server-cluster** commands as shown below:

```
-> vlan 12 members port 1/3 untagged
-> vlan 12 members port 1/4 untagged
-> vlan 12 members port 1/5 untagged
```

```
-> server-cluster 2 port 1/3
-> server-cluster 2 port 1/4
-> server-cluster 2 port 1/5
```

4 Assign an IP address for the by using the **ip interface** command. For example:

```
-> ip interface "vlan 12"
-> ip interface "vlan 12" address 10.135.33.13/24 vlan 12
```

5 Assign mac-address for the new server cluster by using the command **server-cluster mac-address**. For example:

```
-> server-cluster 2 ip 10.135.33.12 mac-address static 01:00:6e:22:33:44
```

Note. *Optional*. You can display the configuration of high availability VLANs with the **show server-cluster** command. For example:

```
-> show server-cluster 2
Cluster Id : 2,
Cluster Name : L3-cluster,
Cluster Mode : L3,
Cluster Mac-address : 01:10:11:22:33:44,
Cluster Vlan : 12,
Administrative State: Enabled,
Operational State : Enabled,
Operational Flag : -
```

An example of what these commands look like entered sequentially on the command line:

```
-> server-cluster 2 mode L3 admin-state enable
-> vlan 12
-> vlan 12 members port 1/3 tagged
-> vlan 12 members port 1/4 tagged
-> vlan 12 members port 1/5 tagged
-> server-cluster 2 port 1/3
-> server-cluster 2 port 1/4
-> server-cluster 2 port 1/5
-> ip interface "vlan 12"
-> ip interface "vlan 12" address 10.135.33.13/24 vlan 12
-> server-cluster 2 ip 10.135.33.12 mac-address static 01:00:6e:22:33:44
```

Example 3: Layer 3 Server Cluster with IP Multicast Address to Cluster (IGMP)

This example shows that a server cluster can be configured with a unique IP address and a IP multicast address. For this scenario, the server cluster IP address needs to be a unicast address and the MAC address (ARP entry) can be unicast or L2 multicast or IP multicast. The MAC address must be configured through CLI ARP resolution to a server cluster MAC, and must be configured before actual routing

.

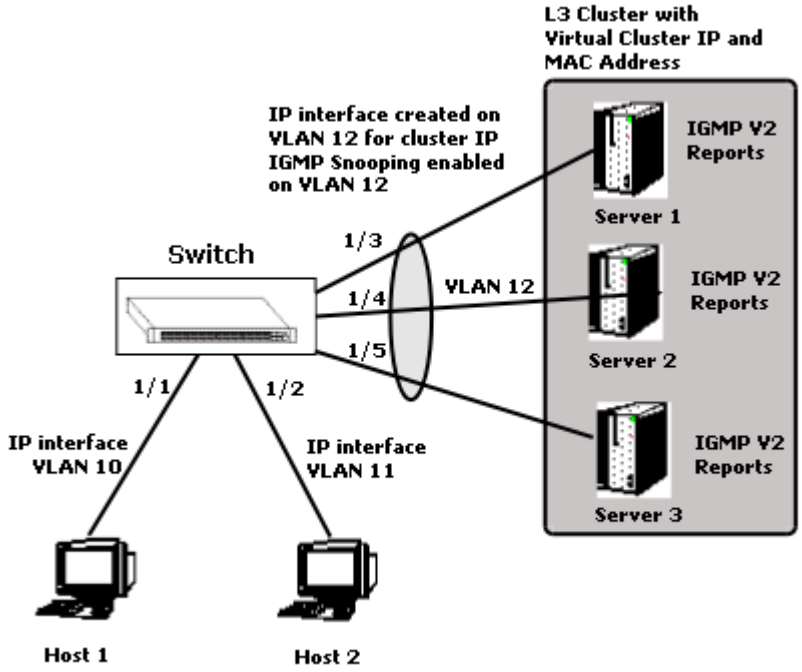


Figure 5-4 : Switch connected to an L3 Server Cluster (IGMP) through 3 ports (1/3,1/4,1/5)

- There is no provision for port list configuration and Ports are derived dynamically using the IGMP snooping of the reports from the server cluster (IGMP v2 reports).
- The traffic which ingresses on 1/1 or 1/2 destined to the server cluster IP is routed to all the ports which are members of the IP multicast group of the server cluster.
- The ingress ports is on a different VLAN as the server cluster IP interface. Join and Leave messaged keep updating the egress port list. However all the egress ports need to be in the same VLAN as the IP interface of server cluster.
- The other traffic is switched according to the normal switching/routing logic.
- Egress port can be a linkagg port as well.

Note. When a server cluster tries to send a bridged or routed packet to itself, a copy of the packet goes back to the sender's (server cluster) port.

Configuration Example

In this example, a packet is an L3 IP switched packet and Egress port can also be a linkagg port.

1 Create a server cluster that will become the HA VLAN by using the command **server-cluster** and configure the mode. For example:

```
-> server-cluster 3 mode L3 admin-state enable
```

2 Create a default VLAN for the HA VLAN ports with the **vlan** command as shown below:

```
-> vlan 12
```

3 Assign member ports to the new default VLAN with the **vlan members untagged** and **server-cluster** commands as shown below:

```
-> vlan 12 members port 1/3 untagged
-> vlan 12 members port 1/4 untagged
-> vlan 12 members port 1/5 untagged
-> server-cluster 3 port 1/3
-> server-cluster 3 port 1/4
-> server-cluster 3 port 1/5
```

4 Assign mac-address for the new server cluster by using the command **server-cluster mac-address**. For example:

```
-> server-cluster 3 ip 10.135.33.12 mac-address static 01:00:11:22:33:44
```

5 If you want to assign a dynamic mac-address for the server cluster, enter the command as follows:

```
-> server-cluster 3 ip 10.135.33.12 mac-address dynamic
```

6 Enable the admin state of the IP multicast by using the **ip multicast admin-state enable** command. IP multicast admin state should be enabled for the IGMP reports to be processed., else the cluster will be operationally down.

```
-> ip multicast admin-state enable
-> server-cluster 3 igmp-mode enable
-> server-cluster 3 ip-multicast 225.0.0.23
```

When IGMP mode is enabled for the server cluster, all static ports will be reset in igmp mode.

Note. *Optional*. You can display the configuration of high availability VLANs with the **show server-cluster** command. For example:

```
-> show server-cluster 3
Cluster Id                : 3,
Cluster Name              : -,
Cluster Mode              : L3,
Cluster IP                : 10.135.33.12,
Cluster Mac-Address       : 01:00:11:22:33:44,
Cluster Mac Type          : Static,
IGMP-Mode                 : Enabled,
Cluster Multicast IP      : 225.0.0.23,
Administrative State      : Enabled,
Operational State         : Disabled,
Operational Flag          : No IGMP members
```

An example of what these commands look like entered sequentially on the command line:

```
-> server-cluster 3 mode L3 admin-state enable
-> vlan 12
-> vlan 12 members port 1/3 untagged
-> vlan 12 members port 1/4 untagged
-> vlan 12 members port 1/5 untagged
-> server-cluster 3 port 1/3
-> server-cluster 3 port 1/4
-> server-cluster 3 port 1/5
-> server-cluster 3 ip 10.135.33.12 mac-address static 01:00:11:22:33:44
-> ip multicast admin-state enable
-> server-cluster 3 igmp-mode enable
-> server-cluster 3 ip-multicast 225.0.0.23
```

Note. In order to process IGMP reports, it is required to enable IP mulitcast by using the **ip multicast admin-state enable** command.

Displaying High Availability VLAN Status

You can use CLI **show** commands to display the current configuration and statistics of high availability VLANs on a switch. These commands include the following:

show server-cluster	Displays the server clusters configured in the system.
show vlan	Displays a list of all VLANs configured on the switch and the status of related VLAN properties (e.g., admin and Spanning Tree status and router port definitions).
show vlan members	Displays a list of VLAN port assignments.

To display the status and configuration of high availability VLANs you use the **show server-cluster** command. To display the status and configuration of all high availability VLANs on a switch, enter the following command:

```
-> show server-cluster
```

A screen similar to the following will be displayed:

```
-> show server-cluster
Legend: * = not valid

Cluster Mode Vlan Mac Address Ip Address IGMP Address Name
---------+-----+-----+------------------+---------------+-------------+-----------
* 10 L2 100 01:10:11:22:33:44     - -    cluster1
  11 L2 100 01:10:11:22:33:44     - -    cluster2
  12 L2 100 01:10:11:22:33:44       - - -
  13 L3 - 01:12:11:22:33:44 10.135.33.203 - -
* 14 L3 - 01:12:11:22:33:45 10.135.33.203 --
  15 L3 - 01:00:6e:00:00:44 10.135.33.203    225.0.1.2 cluster-igmp
```

To display the status and configuration of a single high availability VLAN cluster enter **show server-cluster** followed by the server cluster ID number. For example, to display the status and configuration of high availability server cluster id “1”, enter the following command:

```
-> show server-cluster 1
```

A screen similar to the following will be displayed:

```
-> show server-cluster 1
Cluster Id : 1,
Cluster Name : L2-cluster,
Cluster Mode : L2,
Cluster Mac-address : 01:10:11:22:33:44,
Cluster Vlan : 12,
Administrative State: Enabled,
Operational State : Disabled,
Operational Flag : VPA is not forwarding
```

Note. For more information on the CLI commands, See the *OmniSwitch AOS Release 8 CLI Reference Guide*.

6 Configuring Spanning Tree Parameters

The Spanning Tree Algorithm and Protocol (STP) is a self-configuring algorithm that maintains a loop-free topology on a network. STP helps to provide data path redundancy and network scalability. The OmniSwitch STP implementation, based on the IEEE 802.1D standard, distributes the Spanning Tree load between the primary management module and the network interface modules. This functionality improves network robustness by providing a Spanning Tree that continues to respond to BPDUs (Bridge Protocol Data Unit) and port link up and down states in the event of a fail over to a backup management module or switch.

The OmniSwitch implementation also incorporates the following Spanning Tree features:

- Configures a physical topology into a single Spanning Tree to ensure that there is only one data path between any two switches.
- Supports fault tolerance within the network topology. The Spanning Tree is reconfigured in the event of a data path or bridge failure or when a new switch is added to the topology.
- Supports two Spanning Tree operating modes: *flat* (single STP instance per switch) and *per-VLAN* (single STP instance per VLAN). The per-VLAN mode can be configured to interoperate with the proprietary Per-Vlan Spanning Tree (PVST+) feature of Cisco.
- Supports three Spanning Tree Algorithms; 802.1D (STP), 802.1w (RSTP), and 802.1Q 2005 (MSTP).
- Allows 802.1Q tagged ports and link aggregate logical ports to participate in the calculation of the STP topology.
- Provides loop-guard security to prevent network loops caused due to inconsistencies in data traffic.

The Distributed Spanning Tree software is active on all switches by default. As a result, a loop-free network topology is automatically calculated based on default Spanning Tree bridge, VLAN, and port parameter values. It is only necessary to configure the Spanning Tree parameters to change how the topology is calculated and maintained.

In This Chapter

This chapter provides an overview about how Spanning Tree works and how to configure Spanning Tree parameters through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Selecting the Spanning Tree operating mode (flat or per-VLAN) on page 6-20.
- Configuring Spanning Tree bridge parameters on page 6-26.
- Configuring Spanning Tree port parameters on page 6-33.
- Configuring an example Spanning Tree topology on page 6-43.

Spanning Tree Bridge Parameter Defaults

Parameter Description	Command	Default
Spanning Tree operating mode	**spantree mode**	Per-VLAN (a separate Spanning Tree instance for each VLAN)
PVST+ status	**spantree pvst+compatibility**	Disabled
Spanning Tree status for a VLAN instance	**spantree vlan admin-state**	Enabled
Spanning Tree protocol	**spantree protocol**	RSTP (802.1w)
BPDU switching status	**spantree bpdu-switching**	Disabled
Priority value for the Spanning Tree instance	**spantree priority**	32768
Hello time interval between each BPDU transmission	**spantree hello-time**	2 seconds
Maximum aging time allowed for Spanning Tree information learned from the network	**spantree max-age**	20 seconds
Spanning Tree port state transition time	**spantree forward-delay**	15 seconds
Path cost mode	**spantree path-cost-mode**	Auto (16-bit in per-VLAN mode and STP or RSTP flat mode, 32-bit in MSTP flat mode)
Automatic VLAN Containment	**spantree auto-vlan-containment**	Disabled
Spanning Tree loop-guard	**spantree loop-guard**	Disabled

Spanning Tree Port Parameter Defaults

Parameter Description	Command	Default
Status for a specific VLAN instance	**spantree vlan**	Enabled
Path cost for a specific VLAN instance	**spantree vlan path-cost**	0
Port state management mode	**spantree cist mode** **spantree loop-guard**	Dynamic (Spanning Tree Algorithm determines port state)
Port priority value	**spantree priority**	7
Port connection type for a specific VLAN instance	**spantree vlan connection**	auto point to point
Type of BPDU to be used on a port when per vlan PVST+ mode is enabled	**spantree pvst+compatibility**	auto (IEEE BPDUs are used until a PVST+ BPDU is detected)

Multiple Spanning Tree (MST) Region Defaults

Although the following parameter values are specific to MSTP, they are configurable regardless of which mode (flat or per-VLAN) or protocol is active on the switch.

Parameter Description	Command	Default
The MST region name	**spantree mst region name**	blank
The revision level for the MST region	**spantree mst region revision-level**	0
The maximum number of hops authorized for the region	**spantree mst region max-hops**	20
The number of Multiple Spanning Tree Instances (MSTI)	**spantree msti**	0 (flat mode instance)
The VLAN to MSTI mapping	**spantree msti vlan**	All VLANs are mapped to the Common Internal Spanning Tree (CIST) instance

Spanning Tree Overview

The OmniSwitch supports the use of the 802.1D Spanning Tree Algorithm and Protocol (STP), the 802.1w Rapid Spanning Tree Algorithm and Protocol (RSTP), and the 802.1Q 2005 Multiple Spanning Tree Protocol (MSTP).

RSTP expedites topology changes by allowing blocked ports to transition directly into a forwarding state, bypassing listening and learning states. This provides rapid reconfiguration of the Spanning Tree in the event of a network path or device failure.

The 802.1w standard is an amendment to the 802.1D document, thus RSTP is based on STP. Regardless of which one of these two protocols a switch or VLAN is running, it can successfully interoperate with other switches or VLANs.

802.1Q 2005 is a new version of MSTP that combines the 802.1D 2004 and 802.1S protocols. This implementation of 802.1Q 2005 also includes improvements to edge port configuration and provides administrative control to restrict port role assignment and the propagation of topology change information through bridge ports.

MSTP is an enhancement to the 802.1Q Common Spanning Tree (CST), which is provided when a switch is running in the flat Spanning Tree operating mode. The flat mode applies a single spanning tree instance across all VLAN port connections on a switch. MSTP allows the configuration of Multiple Spanning Tree Instances (MSTIs) in addition to the CST instance. Each MSTI is mapped to a set of VLANs. As a result, the flat mode can now support the forwarding of VLAN traffic over separate data paths.

This section provides a Spanning Tree overview based on RSTP operation and terminology. Although MSTP is based on RSTP, see "MST General Overview" on page 6-12 for specific information about configuring MSTP.

How the Spanning Tree Topology is Calculated

The *tree* consists of links and bridges that provide a single data path that spans the bridged network. At the base of the tree is a *root bridge*. One bridge is elected by all the bridges participating in the network to serve as the root of the tree. After the root bridge is identified, STP calculates the best path that leads from each bridge back to the root and blocks any connections that would cause a network loop.

To determine the best path to the root, STP uses the *path cost* value, which is associated with every port on each bridge in the network. This value is a configurable weighted measure that indicates the contribution of the port connection to the entire path leading from the bridge to the root.

In addition, a *root path cost* value is associated with every bridge. This value is the sum of the path costs for the port that receives frames on the best path to the root (this value is zero for the root bridge). The bridge with the lowest root path cost becomes the *designated bridge* for the LAN, as it provides the shortest path to the root for all bridges connected to the LAN.

During the process of calculating the Spanning Tree topology, each port on every bridge is assigned a *port role* based on how the port and/or its bridge participates in the active Spanning Tree topology.

The following table provides a list of port role types and the port and/or bridge properties that the Spanning Tree Algorithm examines to determine which role to assign to the port.

Role	Port/Bridge Properties
Root Port	Port connection that provides the shortest path (lowest path cost value) to the root. The root bridge does not have a root port.
Designated Port	The designated bridge provides the LAN with the shortest path to the root. The designated port connects the LAN to this bridge.
Backup Port	Any operational port on the designated bridge that is not a root or designated port. Provides a backup connection for the designated port. A backup port can only exist when there are redundant designated port connections to the LAN.
Alternate Port	Any operational port that is not the root port for its bridge and its bridge is not the designated bridge for the LAN. An alternate port offers an alternate path to the root bridge if the root port on its own bridge goes down.
Disabled Port	Port is not operational. If an active connection does come up on the port, it is assigned an appropriate role.

Note. The distinction between a backup port and an alternate port was introduced with the IEEE 802.1w standard to help define rapid transition of an alternate port to a root port.

The role a port plays or can potentially play in the active Spanning Tree topology determines the port operating state; *discarding, learning,* or *forwarding*. The *port state* is also configurable and it is possible to enable or disable the administrative status of a port and/or specify a forwarding or blocking state that is only changed through user intervention.

The Spanning Tree Algorithm only includes ports in its calculations that are operational (link is up) and have an enabled administrative status. The following table compares and defines 802.1D and 802.1w port states and their associated port roles:

STP Port State	RSTP Port State	Port State Definition	Port Role
Disabled	Discarding	Port is down or administratively disabled and is not included in the topology.	Disabled
Blocking	Discarding	Frames are dropped, nothing is learned or forwarded on the port. Port is temporarily excluded from topology.	Alternate, Backup
Learning	Learning	Port is learning MAC addresses that are seen on the port and adding them to the bridge forwarding table, but not transmitting any data. Port is included in the active topology.	Root, Designated
Forwarding	Forwarding	Port is transmitting and receiving data and is included in the active topology.	Root, Designated

Once the Spanning Tree is calculated, there is only one root bridge, one designated bridge for each LAN, and one root port on each bridge (except for the root bridge). Data travels back and forth between bridges over forwarding port connections that form the best, non-redundant path to the root. The active topology ensures that network loops do not exist.

Bridge Protocol Data Units (BPDU)

Switches send layer 2 frames, referred to as Configuration Bridge Protocol Data Units (BPDU), to relay information to other switches. The information in these BPDU is used to calculate and reconfigure the Spanning Tree topology. A Configuration BPDU contains the following information that pertains to the bridge transmitting the BPDU:

Root ID	The Bridge ID for the bridge that this bridge believes is the root.
Root Path Cost	The sum of the Path Costs that lead from the root bridge to this bridge port. The Path Cost is a configurable parameter value. The IEEE 802.1D standard specifies a default value that is based on port speed. See "Configuring Port Path Cost" on page 6-36 for more information.
Bridge ID	An eight-byte hex value that identifies this bridge within the Spanning Tree. The first two bytes contain a configurable priority value and the remaining six bytes contain a bridge MAC address. See "Configuring the Bridge Priority" on page 6-28 for more information. Each switch chassis is assigned a dedicated base MAC address. This is the MAC address that is combined with the priority value to provide a unique Bridge ID for the switch. For more information about the base MAC address, see the appropriate Hardware Users Guide for the switch.
Port ID	A 16-bit hex value that identifies the bridge port that transmitted this BPDU. The first 4 bits contain a configurable priority value and the remaining 12 bits contain the physical switch port number. See "Configuring Port Priority" on page 6-35 for more information.

The sending and receiving of Configuration BPDU between switches participating in the bridged network constitute the root bridge election; the best path to the root is determined and then advertised to the rest of the network. BPDU provide enough information for the STP software running on each switch to determine the following:

- Which bridge serves as the root bridge.
- The shortest path between each bridge and the root bridge.
- Which bridge serves as the designated bridge for the LAN.
- Which port on each bridge serves as the root port.
- The port state (forwarding or discarding) for each bridge port based on the role the port plays in the active Spanning Tree topology.

The following events trigger the transmitting and/or processing of BPDU in order to discover and maintain the Spanning Tree topology:

- When a bridge first comes up, it assumes it is the root and starts transmitting Configuration BPDU on all its active ports advertising its own bridge ID as the root bridge ID.
- When a bridge receives BPDU on its root port that contains more attractive information (higher priority parameters and/or lower path costs), it forwards this information on to other LANs to which it is connected for consideration.

- When a bridge receives BPDU on its designated port that contains information that is less attractive (lower priority values and/or higher path costs), it forwards its own information to other LANs to which it is connected for consideration.

STP evaluates BPDU parameter values to select the best BPDU based on the following order of precedence:

1. The lowest root bridge ID (lowest priority value, then lowest MAC address).
2. The best root path cost.
3. If root path costs are equal, the bridge ID of the bridge sending the BPDU.
4. If the previous three values tie, then the port ID (lowest priority value, then lowest port number).

Topology Change Notification

When a topology change occurs, such as when a link goes down or a switch is added to the network, the affected bridge sends a Topology Change Notification (TCN) BPDU to the designated bridge for its LAN. The designated bridge then forwards the TCN to the root bridge. The root then sends out a Configuration BPDU and sets a Topology Change (TC) flag within the BPDU to notify other bridges that there is a change in the configuration information. Once this change is propagated throughout the Spanning Tree network, the root stops sending BPDU with the TC flag set and the Spanning Tree returns to an active, stable topology.

Note. You can restrict the propagation of TCNs on a port. See "Restricting TCN Propagation" on page 6-42 for more information.

Detecting the Source of Topology Changes

The following information and logging mechanisms are available on each switch to help identify the source of topology changes within an active network:

- The port on which the last TCN was received on the local switch. The "Topology Change Port" field of the **show spantree vlan**, **show spantree cist**, and **show spantree msti** commands displays the switch port on which the last TCN was received. This information can be used to track down the switch that triggered the topology change in an active RSTP or MSTP topology (not supported for STP topologies).
- Switch logging entries to identify root port and root bridge changes for all Spanning Tree protocols (STP, RSTP, and MSTP). For example:

  ```
  2014 May 19 15:26:44 U28E_7_12_7 swlogd: stpCmm _TRPt info(5) TRAP:newRoot stp=0
  2014 May 19 15:39:54 U28E_7_12_7 swlogd: stpCmm _TRPt info(5) TRAP:newRootPort
  stp=0 port=101005
  ```

  For more information about the switch logging utility, see Chapter 38, "Using Switch Logging."

- Topology change storm detection to identify excessive topology changes for all Spanning Tree protocols (STP, RSTP, and MSTP). The switch uses internal calculations based on the number of topology changes within a specific period of time to determine if the number of topology changes exceeds a specific threshold. When this threshold value is reached, switch logging entries are triggered as a warning of potential instability within the network. For example:

  ```
  For Flat + MSTP CIST instance:
  2014 May 19 15:26:44 U28E_7_12_7 swlogd: stpCmm _STPt warn(4) TCN Storm detected
  on port 1/1/1 for Cist
  ```

```
For Flat + MSTP MSTI instance
2014 May 19 15:26:44 U28E_7_12_7 swlogd: stpCmm _STPt warn(4) TCN Storm detected
on port 0/10 for Msti 1001

For Flat + RSTP instance:
2014 May 19 15:26:44 U28E_7_12_7 swlogd: stpCmm _STPt warn(4) TCN Storm detected
on port 1/1/1

For Per VLAN + RSTP instance
2014 May 19 15:26:44 U28E_7_12_7 swlogd: stpCmm _STPt warn(4) TCN Storm detected
on port 1/1/1 for VLAN 1001
```

For more information about the switch logging utility, see Chapter 38, “Using Switch Logging.”

Loop-guard on OmniSwitch

STP relies on continuous reception or transmission of BPDUs based on the port role. The designated port or primary port transmits BPDUs, and the non-designated ports receive BPDUs. When one of the non-designated ports in a spanning tree network stop receiving BPDUs, then the STP conceives that the network is loop free. However, when a non-designated (**Alternate**, **Root**, or **Backup**) port becomes designated and moves to a forwarding state, a loop is created in the network.

With Loop Guard, if a switch stops receiving BPDUs on a non-designated port, the switch places the port into the STP loop-inconsistent blocking state thus preventing the occurrence of loop in the network.

Loop-guard can be configured on individual ports on per-port or per-VLAN basis. A port can have both roles:

- Designated
- Non-designated for mutually exclusive set of VLANs or MSTP-instances (in MSTP mode)

If loop-guard is enabled on the port, it does not affect the forward or blocking state for a designated port. In case of BPDU timeout, if a loop-guard enabled port fails to receive three consecutive BPDUs, STP converts the port explicitly to a blocked port.

When loop-guard is enabled, if a switch stops receiving BPDUs on a non-designated port, the switch places the port into the STP loop-inconsistent blocking state.

By default, loop-guard is disabled on all the switch ports. User can configure loop-guard on any port irrespective of its STP state or role. However, the feature functions only on non-designated (Alternate, Root, or Backup) STP ports.

Notes:

- In the **flat** mode, as there is a single STP instance on all the VLANs, the loop-guard state of the ports is same across all the VLANs on the switch.
- In **MSTP** mode, there is a single STP instance for each MSTI instance. In this case, loop-guard state of port is same across all the VLANs of a given MSTP instance. Hence if a loop-guard error occurs on any single port, it affects all the other ports related to the MSTI.
- In **1X1** mode, there is a single STP instance assigned for each VLAN. Hence if a loop-guard error occurs on any single VLAN, it does not affect the other VLANs.

Topology Examples

The following diagram shows an example of a physical network topology that incorporates data path redundancy to ensure fault tolerance. These redundant paths, however, create loops in the network configuration. If a device connected to Switch A sends broadcast packets, Switch A floods the packets out all of its active ports. The switches connected to Switch A in turn floods the broadcast packets out their active ports, and Switch A eventually receives the same packets back and the cycle starts over again. This causes severe congestion on the network, often referred to as a *broadcast storm*.

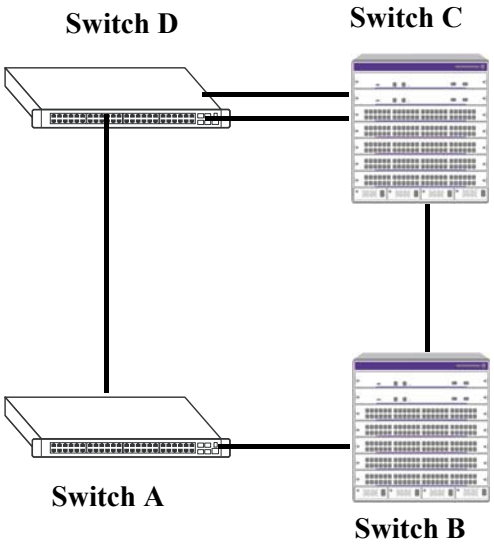


Figure 6-1 : Physical Topology Example

The Spanning Tree Algorithm prevents network loops by ensuring that there is always only one active link between any two switches. This is done by transitioning one of the redundant links into a blocking state, leaving only one link actively forwarding traffic. If the active link goes down, then the Spanning Tree will transition one of the blocked links to the forwarding state to take over for the downed link. If a new switch is added to the network, the Spanning Tree topology is automatically recalculated to include the monitoring of links to the new switch.

The following diagram shows the logical connectivity of the same physical topology as determined by the Spanning Tree Algorithm:

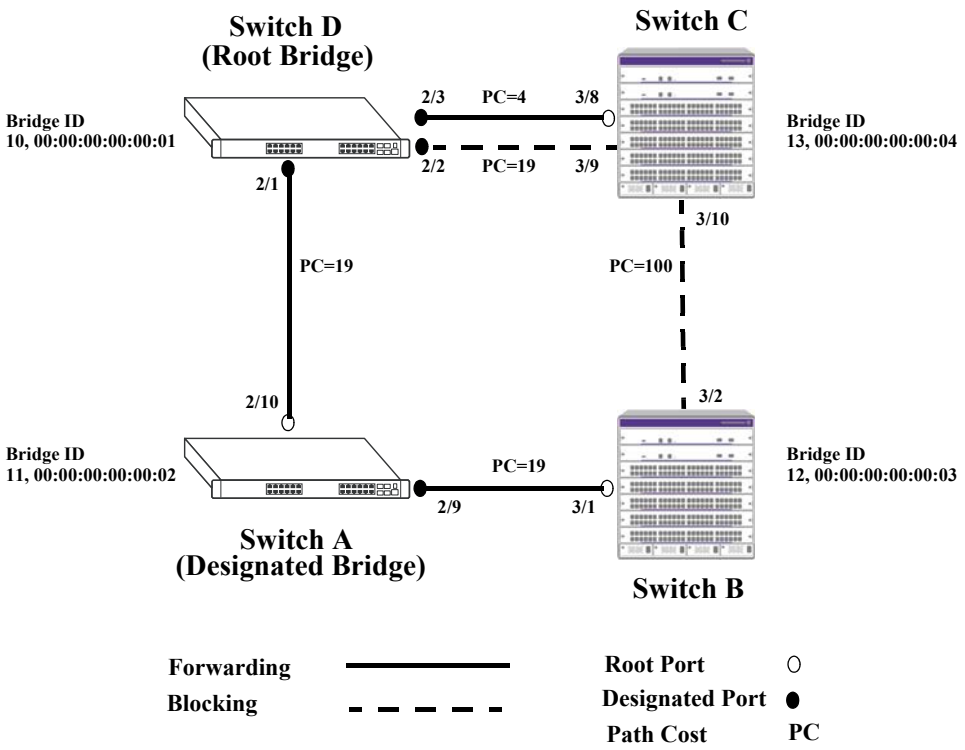


Figure 6-2 : Active Spanning Tree Topology Example

In the above active Spanning Tree topology example, the following configuration decisions were made as a result of calculations performed by the Spanning Tree Algorithm:

- Switch D is the root bridge because its bridge ID has a priority value of 10 (the lower the priority value, the higher the priority the bridge has in the Spanning Tree). If all four switches had the same priority, then the switch with the lowest MAC address in its bridge ID would become the root.
- Switch A is the designated bridge for Switch B, because it provides the best path for Switch B to the root bridge.
- Port 2/9 on Switch A is a designated port, because it connects the LAN from Switch B to Switch A.
- All ports on Switch D are designated ports, because Switch D is the root and each port connects to a LAN.
- Ports 2/10, 3/1, and 3/8 are the root ports for Switches A, B, and C, respectively, because they offer the shortest path towards the root bridge.
- The port 3/9 connection on Switch C to port 2/2 on Switch D is in a discarding (blocking) state, as the connection these ports provides is redundant (backup) and has a higher path cost value than the 2/3 to 3/8 connection between the same two switches. As a result, a network loop is avoided.
- The port 3/2 connection on Switch B to port 3/10 on Switch C is also in a discarding (blocking) state, as the connection these ports provides has a higher path cost to root Switch D than the path between Switch B and Switch A. As a result, a network loop is avoided.

MST General Overview

The Multiple Spanning Tree (MST) feature allows for the mapping of one or more VLANs to a single Spanning Tree instance, referred to as a Multiple Spanning Tree Instance (MSTI), when the switch is running in the flat Spanning Tree mode. MST uses the Multiple Spanning Tree Algorithm and Protocol (MSTP) to define the Spanning Tree path for each MSTI. In addition, MSTP provides the ability to group switches into MST Regions. An MST Region appears as a single, flat Spanning Tree instance to switches outside the region.

This section provides an overview of the MST feature that includes the following topics:

- "How MSTP Works" on page 6-12.
- "Comparing MSTP with STP and RSTP" on page 6-15.
- "What is a Multiple Spanning Tree Instance (MSTI)" on page 6-15.
- "What is a Multiple Spanning Tree Region" on page 6-16.
- "What is the Internal Spanning Tree (IST) Instance" on page 6-17.
- "What is the Common and Internal Spanning Tree Instance" on page 6-17.
- "MST Configuration Overview" on page 6-17.

How MSTP Works

MSTP, as defined in the IEEE 802.1Q 2005 standard, is an enhancement to the IEEE 802.1Q Common Spanning Tree (CST). The CST is a single spanning tree that uses 802.1D (STP) or 802.1w (RSTP) to provide a loop-free network topology.

The OmniSwitch flat spanning tree mode applies a single CST instance on a per switch basis. The per-VLAN mode is an OmniSwitch proprietary implementation that applies a single spanning tree instance on a per VLAN basis. MSTP is only supported in the flat mode and allows for the configuration of additional Spanning Tree instances instead of just the one CST.

On an MSTP flat mode OmniSwitch, the CST is represented by the Common and Internal Spanning Tree (CIST) instance 0 and exists on all switches. Up to 17 instances, including the CIST, are supported. Each additional instance created is referred to as a Multiple Spanning Tree Instance (MSTI). An MSTI represents a configurable association between a single Spanning Tree instance and a set of VLANs.

Note. Although MSTP provides the ability to define MSTIs while running in the flat mode, port state and role computations are automatically calculated by the CST algorithm across all MSTIs. However, it is possible to configure the priority and/or path cost of a port for a particular MSTI so that a port remains in a forwarding state for an MSTI instance, even if it is blocked as a result of automatic CST computations for other instances.

The following diagrams help to further explain how MSTP works by comparing how port states are determined on per-VLAN STP/RSTP mode, flat mode STP/RSTP, and flat mode MSTP switches.

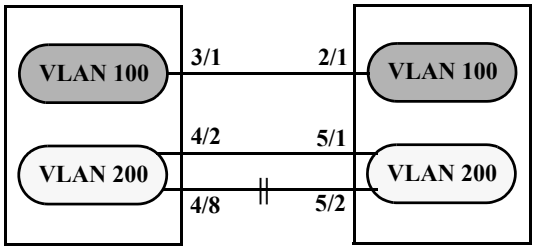


Figure 6-3 : Per-VLAN Mode STP/RSTP

In the above per-VLAN mode example:

- Both switches are running in the per-VLAN mode (one Spanning Tree instance per VLAN).
- VLAN 100 and VLAN 200 are each associated with their own Spanning Tree instance.
- The connection between 3/1 and 2/1 is left in a forwarding state because it is part of the VLAN 100 Spanning Tree instance and is the only connection for that instance.

Note. If additional switches containing a VLAN 100 are connected to the switches in this diagram, then the 3/1 to 2/1 port connection gets into blocking state. The port connection is converted to blocking state, only if the VLAN 100 Spanning Tree instance determines it is required, to avoid a network loop.

- The connections between 4/8 and 5/2 and 4/2 and 5/1 are seen as redundant because they are both controlled by the VLAN 200 Spanning Tree instance and connect to the same switches. The VLAN 200 Spanning Tree instance determines which connection provides the best data path and transitions the other connection to a blocking state.

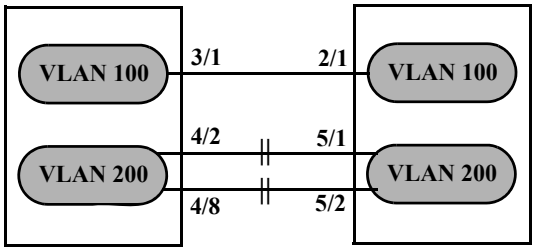


Figure 6-4 : Flat Mode STP/RSTP (802.1D/802.1w)

In the above flat mode STP/RSTP example:

- Both switches are running in the flat mode. As a result, a single flat mode Spanning Tree instance applies to the entire switch and compares port connections across VLANs to determine which connection provides the best data path.
- The connection between 3/1 and 2/1 is left forwarding because the flat mode instance determined that this connection provides the best data path between the two switches.
- The 4/8 to 5/2 connection and the 4/2 to 5/1 connection are considered redundant connections so they are both blocked in favor of the 3/1 to 2/1 connection.

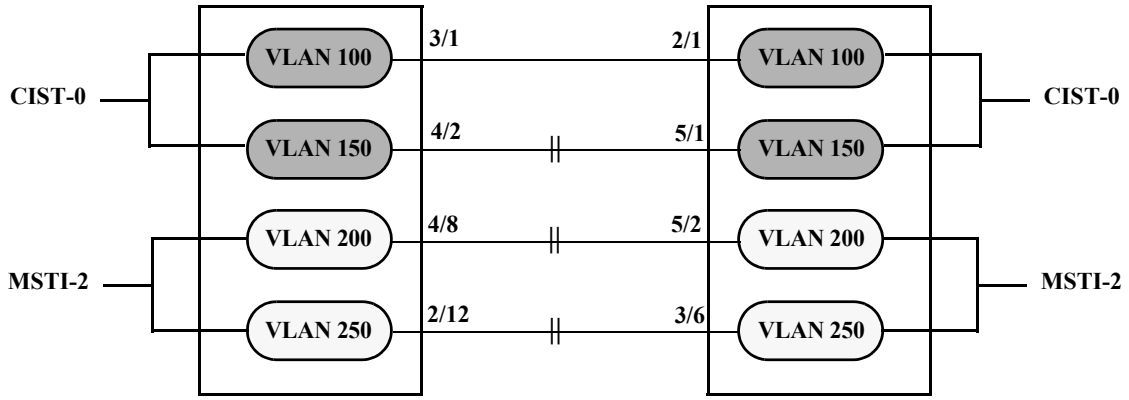


Figure 6-5 : Flat Mode MSTP

In the above flat mode MSTP example:

- Both switches are running in the flat mode and using MSTP.
- VLANs 100 and 150 are *not* associated with an MSTI. They are controlled by the default CIST instance 0 that exists on every switch.
- VLANs 200 and 250 are associated with MSTI 2 so their traffic can traverse a path different from that determined by the CIST.
- Ports are blocked the same way they were blocked in the flat mode STP/RSTP example; all port connections are compared to each other across VLANs to determine which connection provides the best path.

  However, because VLANs 200 and 250 are associated to MSTI 2, it is possible to change the port path cost for ports 2/12, 3/6, 4/8 and/or 5/2 so that they provide the best path for MSTI 2 VLANs, but do not carry CIST VLAN traffic or cause CIST ports to transition to a blocking state.

  Another alternative is to assign all VLANs to an MSTI, leaving no VLANs controlled by the CIST. As a result, the CIST BPDU contains only MSTI information.

See "Sample MSTI Configuration" on page 6-48 for more information about how to direct VLAN traffic over separate data paths using MSTP.

Comparing MSTP with STP and RSTP

Using MSTP has the following items in common with STP (802.1D) and RSTP (802.1w) protocols:

- Each protocol ensures one data path between any two switches within the network topology. This prevents network loops from occurring while at the same time allowing for redundant path configuration.
- Each protocol provides automatic reconfiguration of the network Spanning Tree topology in the event of a connection failure and/or when a switch is added to or removed from the network.
- All three protocols are supported in the flat Spanning Tree operating mode.
- The flat mode CST instance automatically determines port states and roles across VLAN port and MSTI associations. This is because the CST instance is active on all ports and only one BPDU is used to forward information for all MSTIs.
- MSTP is based on RSTP.

Using MSTP differs from STP and RSTP as follows:

- MSTP is only supported when the switch is running in the flat Spanning Tree mode. STP and RSTP are supported in both the per-VLAN and flat modes.
- MSTP allows for the configuration of up to 16 Multiple Spanning Tree Instances (MSTI) in addition to the CST instance. Flat mode STP and RSTP protocols only use the single CST instance for the entire switch. See "What is a Multiple Spanning Tree Instance (MSTI)" on page 6-15 for more information.
- MSTP applies a single Spanning Tree instance to an MSTI ID number that represents a set of VLANs; a one to many association. STP and RSTP in the flat mode apply one Spanning Tree instance to all VLANs; a one to all association. STP and RSTP in the per-VLAN mode apply a single Spanning Tree instance to each existing VLAN; a one to one association.
- The port priority and path cost parameters are configurable for an individual MSTI that represents the VLAN associated with the port.
- The flat mode 802.1D or 802.1w CST is identified as instance 1. When using MSTP, the CST is identified as CIST (Common and Internal Spanning Tree) instance 0. See "What is the Common and Internal Spanning Tree Instance" on page 6-17 for more information.
- MSTP allows the segmentation of switches within the network into MST regions. Each region is seen as a single virtual bridge to the rest of the network, even though multiple switches can belong to the one region. See "What is a Multiple Spanning Tree Region" on page 6-16 for more information.
- MSTP has lower overhead than a per-VLAN configuration. In per-VLAN mode, because each VLAN is assigned a separate Spanning Tree instance, BPDUs are forwarded on the network for each VLAN. MSTP only forwards one BPDU for the CST that contains information for all configured MSTI on the switch.

What is a Multiple Spanning Tree Instance (MSTI)

An MSTI is a single Spanning Tree instance that represents a group of VLANs. The OmniSwitch supports up to 16 MSTIs on one switch. This number is in addition to the Common and Internal Spanning Tree (CIST) instance 0, which is also known as MSTI 0. The CIST instance exists on every switch. By default, all VLANs not mapped to an MSTI are associated with the CIST instance. See "What is the Common and Internal Spanning Tree Instance" on page 6-17 for more information.

What is a Multiple Spanning Tree Region

A Multiple Spanning Tree region represents a group of MSTP switches. An MST region appears as a single, flat mode instance to switches outside the region. A switch can belong to only one region at a time. The region a switch belongs to is identified by the following configurable attributes, as defined by MSTP.

- **Region name** – An alphanumeric string up to 32 characters.
- **Region revision level** – A numerical value between 0 and 65535.
- **VLAN to MSTI table** – Generated when VLANs are associated with MSTIs. Identifies the VLAN to MSTI mapping for the switch.

Switches that share the same values for the configuration attributes described above belong to the same region. For example, in the diagram below:

- Switches A, B, and C all belong to the same region because they all are configured with the same region name, revision level, and have the same VLANs mapped to the same MSTI.
- The CST for the entire network sees Switches A, B, and C as one virtual bridge that is running a single Spanning Tree instance. As a result, CST blocks the path between Switch C and Switch E instead of blocking a path between the MST region switches to avoid a network loop.
- The paths between Switch A and Switch C and the redundant path between Switch B and Switch C were blocked as a result of the Internal Spanning Tree (IST) computations for the MST Region. See "What is the Internal Spanning Tree (IST) Instance" on page 6-17 for more information.

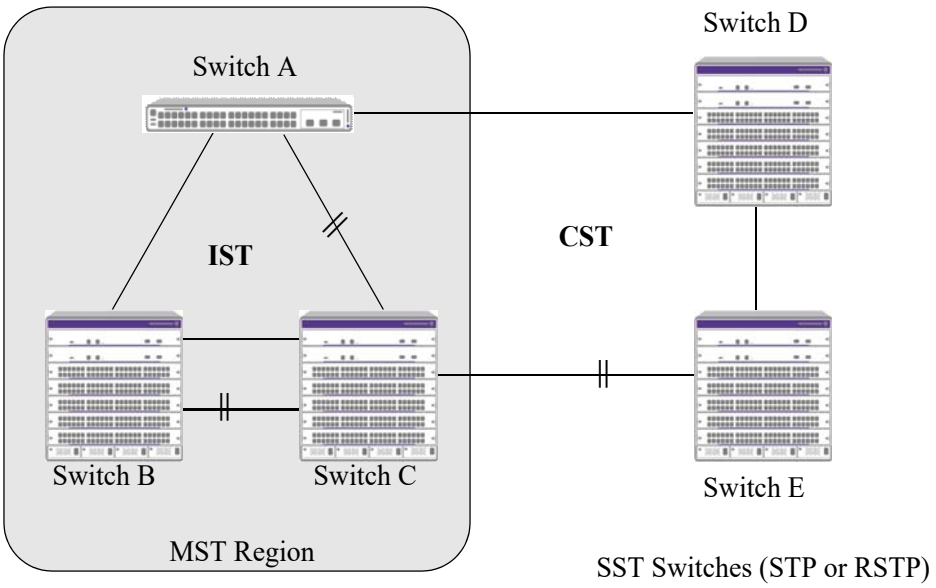


Figure 6-6 : Multiple Spanning Tree region

In addition to the attributes described above, the MST maximum hops parameter defines the number of bridges authorized to propagate MST BPDU information. In essence, this value defines the size of the region in that once the maximum number of hops is reached, the BPDU is discarded.

The maximum number of hops for the region is not one of the attributes that defines membership in the region. See "Sample MST Region Configuration" on page 6-46 for a tutorial on how to configure MST region parameters.

What is the Common Spanning Tree

The Common Spanning Tree (CST) is the overall network Spanning Tree topology resulting from STP, RSTP, and/or MSTP calculations to provide a single data path through the network. The CST provides connectivity between MST regions and other MST regions and/or Single Spanning Tree (SST) switches. For example, in the above diagram, CST calculations detected a network loop created by the connections between Switch D, Switch E, and the MST Region. As a result, one of the paths was blocked.

What is the Internal Spanning Tree (IST) Instance

The IST instance determines and maintains the CST topology between MST switches that belong to the same MST region. In other words, the IST is simply a CST that only applies to MST Region switches while at the same time representing the region as a single Spanning Tree bridge to the network CST.

As shown in the above diagram, the redundant path between Switch B and Switch C is blocked and the path between Switch A and Switch C is blocked. These blocking decisions were based on the IST computations within the MST region. IST sends and receives BPDU to/from the network CST. MSTI within the region do not communicate with the network CST. As a result, the CST only sees the IST BPDU and treats the MST region as a single Spanning Tree bridge.

What is the Common and Internal Spanning Tree Instance

The Common and Internal Spanning Tree (CIST) instance is the Spanning Tree calculated by the MST region IST and the network CST. The CIST is represented by the single Spanning Tree flat mode instance that is available on all switches. By default, all VLANs are associated to the CIST until they are mapped to an MSTI.

When using STP (802.1D) or RSTP (802.1w). When using MSTP, the CIST is also known as instance 0 or MSTI 0.

Note. When MSTP is the active flat mode protocol, explicit Spanning Tree bridge commands are required to configure parameter values. Implicit commands are for configuring parameters when the STP or RSTP protocols are in use. See "Using Spanning Tree Configuration Commands" on page 6-26 for more information.

MST Configuration Overview

The following general steps are required to set up a Multiple Spanning Tree (MST) configuration:

- **Select the flat Spanning Tree mode** – Each switch runs in the default mode. MSTP is only supported on a flat mode switch. See "Spanning Tree Operating Modes" on page 6-20 for more information.
- **Select the MSTP protocol** – Each switch uses the default protocol. Selecting MSTP activates the Multiple Spanning Tree. See "How MSTP Works" on page 6-12 for more information.
- **Configure an MST region name and revision level** – Switches that share the same MST region name, revision level, and VLAN to Multiple Spanning Tree Instance (MSTI) mapping belong to the same MST region. See "What is a Multiple Spanning Tree Region" on page 6-16 for more information.
- **Configure MSTIs** – Every switch has a default Common and Internal Spanning Tree (CIST) instance 0, which is also referred to as MSTI 0. Configuration of additional MSTI is required to segment switch VLANs into separate instances. See "What is a Multiple Spanning Tree Instance (MSTI)" on page 6-15 for more information.

- **Map VLANs to MSTI** – All existing VLANs are mapped to the default CIST instance 0. Associating a VLAN to an MSTI specifies which Spanning Tree instance determines the best data path for traffic carried on the VLAN. In addition, the VLAN-to-MSTI mapping is also one of three MST configuration attributes used to determine that the switch belongs to a particular MST region.

For a tutorial on setting up an example MST configuration, see "Sample MST Region Configuration" on page 6-46 and "Sample MSTI Configuration" on page 6-48.

MST Interoperability and Migration

Connecting an MSTP switch to a non-MSTP flat mode switch is supported. Since the Common and Internal Spanning Tree (CIST) controls the flat mode instance on both switches, STP or RSTP can remain active on the non-MSTP switch within the network topology.

An MSTP switch is part of a Multiple Spanning Tree (MST) Region, which appears as a single, flat mode instance to the non-MSTP switch. The port that connects the MSTP switch to the non-MSTP switch is referred to as a *boundary* port. When a boundary port detects an STP (802.1D) or RSTP (802.1w) BPDU, it responds with the appropriate protocol BPDU to provide interoperability between the two switches. This interoperability also serves to indicate the edge of the MST region.

Interoperability between MSTP switches and per-VLAN mode switches is not recommended. The per-VLAN mode is a proprietary implementation that creates a separate Spanning Tree instance for each VLAN configured on the switch. The MSTP implementation is in compliance with the IEEE standard and is only supported on flat mode switches.

Tagged BPDUs transmitted from a per-VLAN switch are ignored by a flat mode switch. This can cause a network loop to go undetected. Although it is not recommended, you can also connect a per-VLAN switch to a flat mode switch temporarily until migration to MSTP is complete. When a per-VLAN switch is connected to a flat mode switch, configure only a fixed, untagged connection between VLAN 1 on both switches.

Migrating from Flat Mode STP/RSTP to Flat Mode MSTP

Migrating an STP/RSTP flat mode switch to MSTP is relatively transparent. When STP or RSTP is the active protocol, the Common and Internal Spanning Tree (CIST) controls the flat mode instance. If on the same switch the protocol is changed to MSTP, the CIST still controls the flat mode instance.

Note the following when converting a flat mode STP/RSTP switch to MSTP:

- Making a backup copy of the switch **boot.cfg** file before changing the protocol to MSTP is highly recommended. Having a backup copy makes it easier to revert to the non-MSTP configuration. Once MSTP is active, commands are written in their explicit form and not compatible with previous releases of Spanning Tree.
- When converting multiple switches, change the protocol to MSTP first on every switch before starting to configure Multiple Spanning Tree Instances (MSTI).
- Once the protocol is changed, MSTP features are available for configuration. Multiple Spanning Tree Instances (MSTI) are now configurable for defining data paths for VLAN traffic. See "How MSTP Works" on page 6-12 for more information.
- Using explicit Spanning Tree commands to define the MSTP configuration is required. Implicit commands are for configuring STP and RSTP. See "Using Spanning Tree Configuration Commands" on page 6-26 for more information.

- STP and RSTP use a 16-bit port path cost (PPC) and MSTP uses a 32-bit PPC. When the protocol is changed to MSTP, the bridge priority and PPC values for the flat mode CIST instance are reset to their default values.
- It is possible to configure the switch to use 32-bit PPC value for all protocols (see the **spantree path-cost-mode** command page for more information). If this is the case, then the PPC for the CIST is not reset when the protocol is changed to/from MSTP.
- This implementation of MSTP is compliant with the IEEE 802.1Q 2005 standard and thus provides interconnectivity with MSTP compliant systems.

Migrating from Per-VLAN Mode to Flat Mode MSTP

As previously described, the per-VLAN mode is an OmniSwitch proprietary implementation that applies one Spanning Tree instance to each VLAN. For example, if five VLANs exist on the switch, then their are five Spanning Tree instances active on the switch, unless Spanning Tree is disabled on one of the VLANs.

Note the following when converting a per-VLAN mode STP/RSTP switch to flat mode MSTP:

- Making a backup copy of the switch **boot.cfg** file before changing the protocol to MSTP is highly recommended. Having a backup copy makes it easier to revert to the non-MSTP configuration. Once MSTP is active, commands are written in their explicit form and not compatible with previous releases of Spanning Tree.
- Using MSTP requires changing the switch mode from per-VLAN to flat. When the mode is changed from per-VLAN to flat, ports still retain their VLAN associations but are now part of a single, flat mode Spanning Tree instance that spans across all VLANs. As a result, a path that was forwarding traffic in the per-VLAN mode transitions to a blocking state after the mode is changed to flat.
- Once the protocol is changed, MSTP features are available for configuration. Multiple Spanning Tree Instances (MSTI) are now configurable for defining data paths for VLAN traffic. See "How MSTP Works" on page 6-12 for more information.
- Note that STP/RSTP use a 16-bit port path cost (PPC) and MSTP uses a 32-bit PPC. When the protocol is changed to MSTP, the bridge priority and PPC values for the flat mode CIST instance are reset to their default values.
- It is possible to configure the switch to use 32-bit PPC value for all protocols (see the **spantree path-cost-mode** command page for more information). If this is the case, then the PPC for the CIST is not reset when the protocol is changed to/from MSTP.
- This implementation of MSTP is compliant with the IEEE 802.1Q 2005 standard and thus provides interconnectivity with MSTP compliant systems.

Spanning Tree Operating Modes

The switch can operate in one of two Spanning Tree modes: *flat* and *per-VLAN*. Both modes apply to the entire switch and determine whether a single Spanning Tree instance is applied across multiple VLANs (flat mode) or a single instance is applied to each VLAN (per-VLAN mode). A switch runs on the default mode when it is first turned on.

Use the **spantree mode** command to select the Flat or Per-VLAN Spanning Tree mode.The switch operates in one mode or the other, however, it is not necessary to reboot the switch when changing modes.

Using Flat Spanning Tree Mode

Before selecting the flat Spanning Tree mode, consider the following:

- If STP (802.1D) is the active protocol, then there is one Spanning Tree instance for the entire switch; port states are determined across VLANs. If MSTP (802.1s) is the active protocol, then multiple instances up to a total of 17 are allowed. Port states, however, are still determined across VLANs.
- Multiple connections between switches are considered redundant paths even if they are associated with different VLANs.
- Spanning Tree parameters are configured for the single flat mode instance. For example, if Spanning Tree is disabled on VLAN 1, then it is disabled for all VLANs. Disabling STP on any other VLAN, however, only exclude ports associated with that VLAN from the Spanning Tree Algorithm.
- Fixed (untagged) and 802.1Q tagged ports are supported in each VLAN. BPDU, however, are always untagged.
- When the Spanning Tree mode is changed from per-VLAN to flat, ports still retain their VLAN associations but are now part of a single Spanning Tree instance that spans across all VLANs. As a result, a path that was forwarding traffic in the per-VLAN mode can transition to a blocking state after the mode is changed to flat.

To change the Spanning Tree operating mode to flat, enter the following command:

```
-> spantree mode flat
```

The following diagram shows a flat mode switch with STP (802.1D) as the active protocol. All ports, regardless of their default VLAN configuration or tagged VLAN assignments, are considered part of one Spanning Tree instance. To see an example of a flat mode switch with MSTP (802.1s) as the active protocol, see "MST General Overview" on page 6-12.

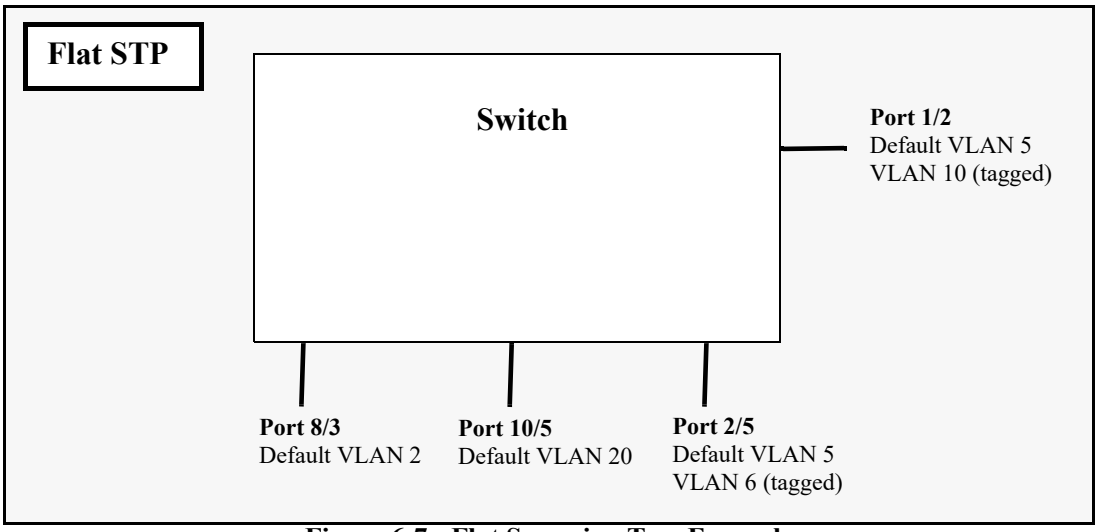


Figure 6-7 : Flat Spanning Tree Example

In the above example, if port 8/3 connects to another switch and port 10/5 connects to that same switch, the Spanning Tree Algorithm would detect a redundant path and transition one of the ports into a blocking state. The same holds true for the tagged ports.

Using Per-VLAN Spanning Tree Mode

Before selecting the Per-VLAN Spanning Tree operating mode, consider the following:

- A single Spanning Tree instance is enabled for each VLAN configured on the switch. For example, if there are five VLANs configured on the switch, then there are five separate Spanning Tree instances, each with its own root VLAN. In essence, a VLAN is a virtual bridge. The VLAN has its own bridge ID and configurable STP parameters, such as protocol, priority, hello time, max age, and forward delay.
- Port state is determined on a per VLAN basis. For example, port connections in VLAN 10 are only examined for redundancy within VLAN 10 across all switches. If a port in VLAN 10 and a port in VLAN 20 both connect to the same switch within their respective VLANs, they are not considered redundant data paths and STP does not block them. However, if two ports within VLAN 10 both connect to the same switch, then the STP transition one of these ports to a blocking state.
- Fixed (untagged) ports participate in the single Spanning Tree instance that applies to their configured default VLAN.
- 802.1Q tagged ports participate in an 802.1Q Spanning Tree instance that allows the Spanning Tree to extend across tagged VLANs. As a result, a tagged port can participate in more than one Spanning Tree instance; one for each VLAN that the port carries.
- If a VLAN contains both fixed and tagged ports, then a hybrid of the two Spanning Tree instances (single and 802.1Q) is applied. If a VLAN appears as a tag on a port, then the BPDU for that VLAN are also tagged. However, if a VLAN appears as the configured default VLAN for the port, then BPDU are not tagged and the single Spanning Tree instance applies.

To change the Spanning Tree operating mode to per-VLAN, enter the following command:

```
-> spantree mode per-vlan
```

The following diagram shows a switch running in the per-VLAN Spanning Tree mode and shows Spanning Tree participation for both fixed and tagged ports.

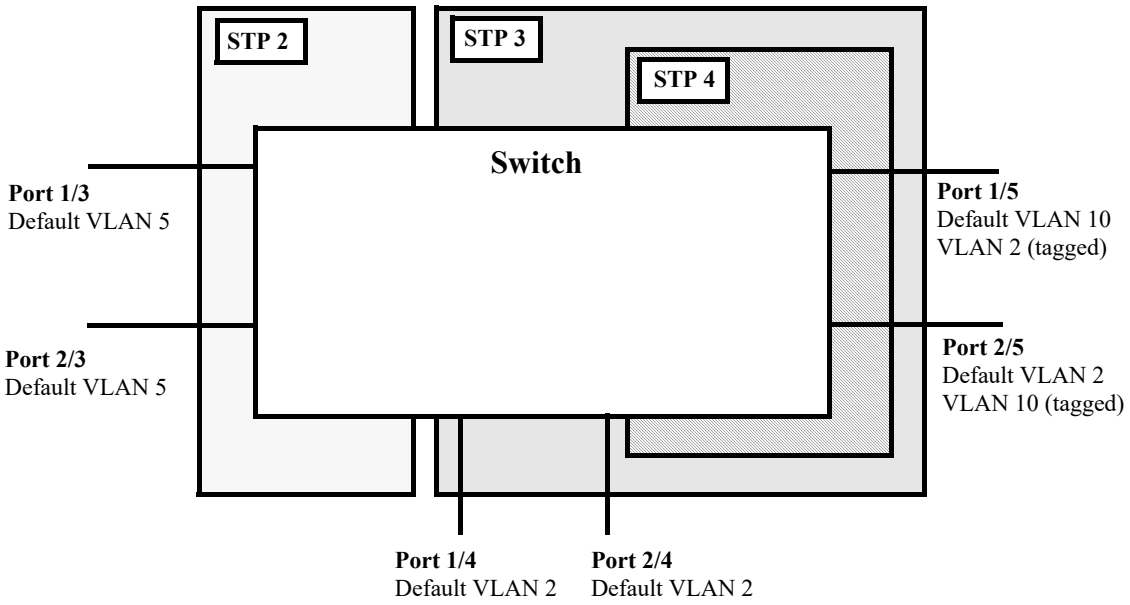


Figure 6-8 : Per VLAN (single and 802.1Q) Spanning Tree Example

In the above example, STP2 is a single Spanning Tree instance since VLAN 5 contains only fixed ports. STP 3 and STP 4 are a combination of single and 802.1Q Spanning Tree instances because VLAN 2 contains both fixed and tagged ports. On ports where VLAN 2 is the default VLAN, BPDU are not tagged. on ports where VLAN 2 is a tagged VLAN, BPDU are also tagged.

Using Per-VLAN Spanning Tree Mode with PVST+

In order to interoperate with Cisco's proprietary Per Vlan Spanning Tree (PVST+) mode, the OmniSwitch per-VLAN Spanning Tree mode allows OmniSwitch ports to transmit and receive either the standard IEEE BPDUs or Cisco's proprietary PVST+ BPDUs. When the PVST+ mode is enabled, a user port operates in the default mode initially until it detects a PVST+ BPDU, which automatically enables the port to operate in the Cisco PVST+ compatible mode.

The PVST+ compatibility mode allows OmniSwitch ports to operate in the per-VLAN mode when connected to another OmniSwitch or in the Cisco PVST+ mode when connected to a Cisco switch. As a result, both the OmniSwitch per-VLAN and Cisco PVST+ modes can co-exist on the same OmniSwitch *and* interoperate correctly with a Cisco switch using the standard Spanning Tree protocols (STP or RSTP).

Note. In the flat Spanning Tree mode, both the OmniSwitch and Cisco switches can interoperate seamlessly using the standard MSTP protocol.

OmniSwitch PVST+ Interoperability

Native VLAN and OmniSwitch Default VLAN

Cisco uses the standard IEEE BPDU format for the native VLAN (VLAN 1) over an 802.1Q trunk. Thus, by default the Common Spanning Tree (CST) instance of the native VLAN 1 for all Cisco switches and the STP instance for the default VLAN of a port on an OmniSwitch interoperates and successfully creates a loop-free topology.

802.1Q Tagged VLANs

For 802.1Q tagged VLANs, Cisco uses a proprietary frame format which differs from the standard IEEE BPDU format used by the OmniSwitch per-VLAN mode, thus preventing Spanning Tree topologies for tagged VLANs from interoperating over the 802.1Q trunk.

In order to interoperate with Cisco PVST+ mode, the current OmniSwitch per-VLAN mode has an option to recognize Cisco's proprietary PVST+ BPDUs. This allows any user port on an OmniSwitch to send and receive PVST+ BPDUs, so that loop-free topologies for the tagged VLANs can be created between OmniSwitch and Cisco switches.

Configuration Overview

The **spantree pvst+compatibility** command is used to enable or disable the PVST+ interoperability mode globally for all switch ports and link aggregates or on a per-port/link aggregate basis. By default, PVST+ compatibility is disabled.

To globally enable or disable PVST+ interoperability, enter the following commands:

```
-> spantree pvst+compatibility enable
-> spantree pvst+compatibility disable
```

To enable or disable PVST+ interoperability for a specific port or link aggregate, use the **spantree pvst+compatibility** command with the **port** or **linkagg** parameter. For example:

```
-> spantree pvst+compatibility port 1/3 enable
-> spantree pvst+compatibility port 2/24 disable
-> spantree pvst+compatibility linkagg 3 enable
-> spantree pvst+compatibility linkagg 10 disable
```

The following causes a port to exit from the enabled state:

- The link status of the port changes.
- The administrative status of the port changes.
- The PVST+ status of the port is disabled or set to auto.

To configure a port or link aggregate to automatically detect

The **spantree pvst+compatibility** command also provides an **auto** option to configure the port to handle IEEE BPDUs initially (i.e., disable state). Once a PVST+ BPDU is received, it handles the PVST+ BPDUs and IEEE BPDUs for a Cisco native VLAN. For example:

```
-> spantree pvst+compatibility port 1/3 auto
-> spantree pvst+compatibility linkagg 3 auto
```

The following show command displays the PVST+ status.

```
-> show spantree mode
Spanning Tree Global Parameters
Current Running Mode  : per-vlan,
Current Protocol      : N/A (Per VLAN),
Path Cost Mode        : 32 BIT,
Auto Vlan Containment : N/A
Cisco PVST+ mode      : Enabled
Vlan Consistency check: Disabled
```

BPDU Processing in PVST+ Mode

An OmniSwitch port operating in PVST+ mode processes BPDUs as follows:

If the default VLAN of a port is VLAN 1 then:

- Send and receive IEEE untagged BPDUs for VLAN 1
- Don't send and receive PVST+ tagged BPDUs for VLAN 1
- Send and receive tagged PVST+ BPDUs for other tagged VLANs.

If the default VLAN of a port is not VLAN 1 then:

- Send and receive IEEE untagged BPDUs for VLAN 1
- Don't send and receive PVST+ tagged BPDUs for VLAN 1
- Send and receive untagged PVST+ BPDUs for the port's default VLAN
- Send and receive tagged PVST+ BPDUs for other tagged VLANs

Recommendations and Requirements for PVST+ Configurations

- It is mandatory that all the Cisco switches have the MAC Reduction Mode feature enabled in order to interoperate with an OmniSwitch in PVST+ mode. This avoids any unexpected election of a root bridge.
- You can assign the priority value only in the multiples of 4096 to be compatible with the Cisco MAC Reduction mode; any other values result in an error message. Also, the existing per vlan priority values are restored when changing from PVST+ mode back to per-VLAN mode. For more information on priority, refer to “Configuring the Bridge Priority” on page 6-28.
- In a mixed OmniSwitch and Cisco environment, it is highly recommended to enable PVST+ mode on all OmniSwitches in order to maintain the same root bridge for the topology. It is possible that the new root bridge might be elected as a result of inconsistencies of MAC reduction mode when connecting an OmniSwitch that does not support Cisco PVST+ mode to an OmniSwitch with the PVST+ mode enabled. In this case, the root bridge priority must be changed manually to maintain the same root bridge. For more information on priority, refer to “Configuring the Bridge Priority” on page 6-28.
- A Cisco switch running in PVST mode (another Cisco proprietary mode prior to 802.1q standard) is not compatible with an OmniSwitch running in per-VLAN PVST+ mode.
- Both Cisco and OmniSwitch support two default path cost modes; long or short. It is recommended that the same default path cost mode be configured in the same way on all switches so that the path costs for similar interface types are consistent when connecting ports between OmniSwitch and Cisco Switches. For more information on path cost mode, refer to “Configuring the Path Cost Mode” on page 6-31.

- Dynamic aggregate link (LACP) functions properly between OmniSwitch and Cisco switches. The Cisco switches send the BPDUs only on one physical link of the aggregate, similar to the OmniSwitch Primary port functionality. The path cost assigned to the aggregate link is not the same between OmniSwitch and Cisco switches since vendor-specific formulas are used to derive the path cost. Manual configuration is recommended to match the Cisco path cost assignment for an aggregate link. For more information on the configuration of path cost for aggregate links, refer to "Path Cost for Link Aggregate Ports" on page 6-38.

The table below shows the default Spanning Tree values.

Parameters	**OmniSwitch**	**Cisco**
Mac Reduction Mode	Enabled	Disabled
Bridge Priority	32768	32768
Port Priority	128	32 (catOS) / 128 (IOS)
Port Path Cost	IEEE Port Speed Table	IEEE Port Speed Table
Aggregate Path Cost	Proprietary Table	Avg Path Cost / NumPorts
Default Path Cost Mode	Short (16-bit)	Short (16-bit)
Max Age	20	20
Hello Time	2	2
Forward Delay Time	15	15
Default Protocol	RSTP (1w) Per Vlan	PVST+ (1d) Per Switch

Using Spanning Tree Configuration Commands

The OmniSwitch Spanning Tree implementation uses commands that contain one of the following keywords to specify the type of Spanning Tree instance to modify:

- **cist** – command applies to the Common and Internal Spanning Tree instance. The CIST is the single Spanning Tree flat mode instance that is available on all switches. When using STP or RSTP, the CIST is also known as instance 1 or bridge 1.
- **msti** – command applies to the specified Multiple Spanning Tree Instance. When using MSTP (802.1s), the CIST instance is also known as MSTI 0.
- **vlan** – command applies to the specified VLAN instance.

These commands (referred to as explicit commands) allow the configuration of a particular Spanning Tree instance independent of which mode and/or protocol is currently active on the switch. The configuration, however, does not go active until the switch is changed to the appropriate mode. For example, if the switch is running in the per-VLAN mode, the following explicit command changes the MSTI 3 priority to 12288:

```
-> spantree msti 3 priority 12288
```

Even though the above command is accepted in the per-VLAN mode, the new priority value does not take effect until the switch mode is changed to flat mode.

Note. When a snapshot is taken of the switch configuration, the explicit form of all Spanning Tree commands is captured. For example, if the priority of MSTI 2 was changed from the default value to a priority of 16384, then **spantree msti 2 priority 16384** is the command captured to reflect this in the snapshot file. In addition, explicit commands are captured for both flat and per-VLAN mode configurations.

Configuring STP Bridge Parameters

The Spanning Tree software is active on all switches by default and uses default bridge and port parameter values to calculate a loop free topology. It is only necessary to configure these parameter values if it is necessary to change how the topology is calculated and maintained.

Note the following when configuring Spanning Tree bridge parameters:

- When a switch is running in the per-VLAN Spanning Tree mode, each VLAN is in essence a virtual bridge with its own Spanning Tree instance and configurable bridge parameters.
- When the switch is running in the flat mode and STP (802.1D) or RSTP (802.1w) is the active protocol, bridge parameter values are only configured for the flat mode instance.
- If MSTP (802.1s) is the active protocol, then the priority value is configurable for each Multiple Spanning Tree Instance (MSTI). All other parameters, however, are still only configured for the flat mode instance and are applied across all MSTIs.
- Bridge parameter values for a VLAN instance are not active unless Spanning Tree is enabled on the VLAN and at least one active port is assigned to the VLAN. Use the **spantree vlan admin-state** command to enable or disable a VLAN Spanning Tree instance.

- If Spanning Tree is disabled on a VLAN, active ports associated with that VLAN are excluded from Spanning Tree calculations and remain in a forwarding state.
- Note that when a switch is running in the flat mode, disabling Spanning Tree on VLAN 1 disables the instance for all VLANs and all active ports are then excluded from any Spanning Tree calculations and remain in a forwarding state.

The following is a summary of Spanning Tree bridge configuration commands. For more information about these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide.*

Commands	Used for ...
spantree protocol	Configuring the protocol for the flat mode CIST instance or a per-VLAN mode VLAN instance.
spantree priority	Configuring the priority value for the flat mode CIST instance, a Multiple Spanning Tree Instance (MSTI), or a per-VLAN mode VLAN instance.
spantree hello-time	Configuring the hello time value for the flat mode CIST instance or a per-VLAN mode VLAN instance.
spantree max-age	Configuring the maximum age time value for the flat mode CIST instance or a per-VLAN mode VLAN instance.
spantree forward-delay	Configuring the forward delay time value for the flat mode CIST instance or a per-VLAN mode VLAN instance.
spantree bpdu-switching	Configuring the BPDU switching status for a VLAN.
spantree path-cost-mode	Configuring the automatic selection of a 16-bit path cost for STP/RSTP ports and a 32-bit path cost for MSTP ports or sets all path costs to use a 32-bit value.
spantree auto-vlan-containment	Enables or disables Auto VLAN Containment (AVC) for 802.1s instances.
spantree pvst+compatibility	Enables or disables PVST+ mode on the switch.

The following sections provide information and procedures for using the bridge configuration commands and also includes command examples.

Selecting the Spantree Protocol

The switch supports three Spanning Tree protocols: STP, RSTP (the default), MSTP. To configure the Spanning Tree protocol for a VLAN instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree protocol** command with the **vlan** parameter. For example, the following command changes the protocol to RSTP for VLAN 455:

```
-> spantree vlan 455 protocol rstp
```

Note. When configuring the protocol value for a VLAN instance, MSTP is not an available option. This protocol is only supported on the flat mode instance.

To configure the protocol for the flat mode CIST instance, use either the **spantree protocol** command or the **spantree protocol** command with the **cist** parameter. Note that both commands are available when the switch is running in either mode (per-VLAN or flat). For example, the following commands configure the protocol for the flat mode instance to MSTP:

```
-> spantree cist protocol mstp
-> spantree protocol mstp
```

Configuring the Bridge Priority

A bridge is identified within the Spanning Tree by its bridge ID (an eight byte hex number). The first two bytes of the bridge ID contain a priority value and the remaining six bytes contain a bridge MAC address.

The bridge priority is used to determine which bridge serves as the root of the Spanning Tree. The lower the priority value, the higher the priority. If more than one bridge have the same priority, then the bridge with the lowest MAC address becomes the root.

Note. Configuring a Spanning Tree bridge instance with a priority value that causes the instance to become the root is recommended, instead of relying on the comparison of switch base MAC addresses to determine the root.

If the switch is running in the per-VLAN Spanning Tree mode, then a priority value is assigned to each VLAN instance. If the switch is running in the flat Spanning Tree mode, the priority is assigned to the flat mode instance or a Multiple Spanning Tree Instance (MSTI). In both cases, the default priority value is assigned. Note that priority value for an MSTI must be a multiple of 4096.

To change the bridge priority value for a VLAN instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree priority** command with the **vlan** parameter. For example, the following command changes the priority for VLAN 455 to 25590:

```
-> spantree vlan 455 priority 25590
```

Note. If PVST+ mode is enabled on the switch, then the priority values can be assigned only in the multiples of 4096 to be compatible with the Cisco MAC Reduction mode; any other values result in an error message.

To change the bridge priority value for the flat mode CIST instance, use either the **spantree priority** command or the **spantree priority** command with the **cist** parameter. Note that both commands are available when the switch is running in either mode (per-VLAN or flat). For example, the following commands change the bridge priority value for the flat mode instance to 12288:

```
-> spantree cist priority 12288
-> spantree priority 12288
```

The bridge priority value is also configurable for a Multiple Spanning Tree Instance (MSTI). To configure this value for an MSTI, use the **spantree priority** command with the **msti** parameter and specify a priority value that is a multiple of 4096. For example, the following command configures the priority value for MSTI 10 to 61440:

```
-> spantree msti 10 priority 61440
```

Configuring the Bridge Hello Time

The bridge hello time interval is the number of seconds a bridge waits between transmissions of Configuration BPDU. When a bridge is attempting to become the root or if it has become the root or a designated bridge, it sends Configuration BPDU out all forwarding ports once every hello time value.

The hello time propagated in a root bridge Configuration BPDU is the value used by all other bridges in the tree for their own hello time. Therefore, if this value is changed for the root bridge, all other bridges associated with the same STP instance adopt this value as well.

Note. Lowering the hello time interval improves the robustness of the Spanning Tree algorithm. Increasing the hello time interval lowers the overhead of Spanning Tree processing.

If the switch is running in the per-VLAN Spanning Tree mode, then a hello time value is defined for each VLAN instance. If the switch is running in the flat Spanning Tree mode, then a hello time value is defined for the single flat mode instance. In both cases, the default hello time value is used.

To change the bridge hello time value for a VLAN instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree hello-time** command with the **vlan** parameter. For example, the following command changes the hello time for VLAN 455 to 5 seconds:

```
-> spantree vlan 455 hello-time 5
```

To change the bridge hello time value for the flat mode CIST instance, use either the **spantree hello-time** command or the **spantree hello-time** command with the **cist** parameter. Note that both commands are available when the switch is running in either mode (per-VLAN or flat). For example, the following commands change the hello time value for the flat mode instance to 10:

```
-> spantree hello-time 10
-> spantree cist hello-time 10
```

Note that the bridge hello time is not configurable for Multiple Spanning Tree Instances (MSTI). These instances inherit the hello time from the flat mode instance (CIST).

Configuring the Bridge Max-Age Time

The bridge max-age time specifies how long, in seconds, the bridge retains Spanning Tree information it receives from Configuration BPDU. When a bridge receives a BPDU, it updates its configuration information and the max age timer is reset. If the max age timer expires before the next BPDU is received, the bridge attempts to become the root, designated bridge, or change its root port.

The max-age time propagated in a root bridge Configuration BPDU is the value used by all other bridges in the tree for their own max-age time. Therefore, if this value is changed for the root bridge, all other VLANs associated with the same instance adopt this value as well.

If the switch is running in the per-VLAN Spanning Tree mode, then a max-age time value is defined for each VLAN instance. If the switch is running in the flat Spanning Tree mode, then the max-age value is defined for the flat mode instance. In both cases, the default max-age time is used.

Note. Configuring a low max-age time can cause Spanning Tree to reconfigure the topology more often.

To change the bridge max-age time value for a VLAN instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree max-age** command with the **vlan** parameter. For example, the following command changes the max-age time for VLAN 455 to 10 seconds:

```
-> spantree vlan 455 max-age 10
```

To change the max-age time value for the flat mode CIST instance, use either the **spantree max-age** command or the **spantree max-age** command with the **cist** parameter. Note that both commands are available when the switch is running in either mode (per-VLAN or flat). For example, the following commands change the max-age time value for the flat mode instance to 10:

```
-> spantree max-age 10
-> spantree cist max-age 10
```

Note. The max-age time is not configurable for Multiple Spanning Tree Instances (MSTI). These instances inherit the max-age time from the flat mode instance (CIST).

Configuring the Forward Delay Time for the Switch

The bridge forward delay time specifies how long, in seconds, a port remains in the learning state while it is transitioning to a forwarding state. In addition, when a topology change occurs, the forward delay time value is used to age out all dynamically learned addresses in the MAC address forwarding table. For more information about the MAC address table, see Chapter 3, "Managing Source Learning."

The forward delay time propagated in a root bridge Configuration BPDU is the value used by all other bridges in the tree for their own forward delay time. Therefore, if this value is changed for the root bridge, all other bridges associated with the same instance adopt this value as well.

If the switch is running in the per-VLAN Spanning Tree mode, then a forward delay time value is defined for each VLAN instance. If the switch is running in the flat Spanning Tree mode, then the forward delay time value is defined for the flat mode instance. In both cases, the default forward delay time is used.

Note. Specifying a low forward delay time can cause temporary network loops, because packets can get forwarded before Spanning Tree configuration or change notices have reached all nodes in the network.

To change the bridge forward delay time value for a VLAN instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree forward-delay** command with the vlan **parameter**. For example, the following command changes the forward delay time for VLAN 455 to 10 seconds:

```
-> spantree vlan 455 forward-delay 10
```

To change the forward-delay time value for the flat mode CIST instance, use either the **spantree forward-delay** command or the **spantree forward-delay** command with the **cist** parameter. Note that both commands are available when the switch is running in either mode (per-VLAN or flat). For example, the following commands change the forward-delay time value for the flat mode instance to 10:

```
-> spantree forward-delay 10
-> spantree cist forward-delay 10
```

Note. The forward delay time is not configurable for Multiple Spanning Tree Instances (MSTI). These instances inherit the forward delay time from the flat mode instance (CIST).

Enabling/Disabling the VLAN BPDU Switching Status

BPDU are not switched on ports associated with VLANs that have Spanning Tree disabled. This can result in a network loop if the VLAN has redundant paths to one or more other switches. Allowing VLANs that have Spanning Tree disabled to forward BPDU to all ports in the VLAN, can help to avoid this problem.

To enable or disable the switching of Spanning Tree BPDU for all VLAN and CIST instances when the switch is running in the per-VLAN mode, use the **spantree bpdu-switching** command:

```
-> spantree bpdu-switching enable
-> spantree bpdu-switching disable
```

To enable or disable the switching of Spanning Tree BPDU for only the CIST instance when the switch is running in the flat mode, use the **spantree bpdu-switching** command:

```
-> spantree cist bpdu-switching enable
-> spantree cist bpdu-switching disable
```

To enable or disable BPDU switching on a VLAN, use the **vlan** parameter along with **spantree bpdu-switching** command. For example, the following commands enable BPDU switching on VLAN 10 and disable it on VLAN 20:

```
-> spantree vlan 10 bpdu-switching enable
-> spantree vlan 20 bpdu-switching disable
```

Note. Disabling BPDU switching on a Spanning Tree disabled VLAN must not cause network loops to go undetected.

Configuring the Path Cost Mode

The path cost mode controls whether the switch uses a 16-bit port path cost (PPC) or a 32-bit PPC. When a 32-bit PPC switch connects to a 16-bit PPC switch, the 32-bit switch has a higher PPC value that advertises an inferior path cost to the 16-bit switch. In this case, it is desirable to set the 32-bit switch to use STP or RSTP with a 16-bit PPC value.

The path cost mode is automatically set to use a 16-bit value for all ports that are associated with an STP instance or an RSTP instance and a 32-bit value for all ports associated with an MSTP value. It is also possible to set the path cost mode to always use a 32-bit regardless of which protocol is active.

To change the path cost mode, use the **spantree path-cost-mode** command and specify either **auto** (uses PPC value based on protocol) or **32bit** (always use a 32-bit PPC value). For example, the following command changes the default path cost mode from auto to 32-bit:

```
-> spantree path-cost-mode 32bit
```

Note. Cisco supports two default path cost modes: long or short just like in OmniSwitch per-VLAN implementation. If you have configured PVST+ mode in the OmniSwitch, it is recommended that the same default path cost mode must be configured in the same way in all the switches, so that, the path costs for similar interface types are consistent when connecting ports between OmniSwitch and Cisco switches.

Using Automatic VLAN Containment

In a Multiple Spanning Tree (MST) configuration, it is possible for a port that belongs to a VLAN that is not a member of an instance to become the root port for that instance. This can cause a topology change that could lead to a loss of connectivity between VLANs/switches. Enabling Automatic VLAN Containment (AVC) helps to prevent this from happening by making such a port an undesirable choice for the root.

When AVC is enabled, it identifies undesirable ports and automatically configures them with an infinite path cost value. For example, in the following diagram a link exists between VLAN 2 on two different switches. The ports that provide this link belong to default VLAN 1 but are tagged with VLAN 2. In addition, VLAN 2 is mapped to MSTI 1 on both switches.

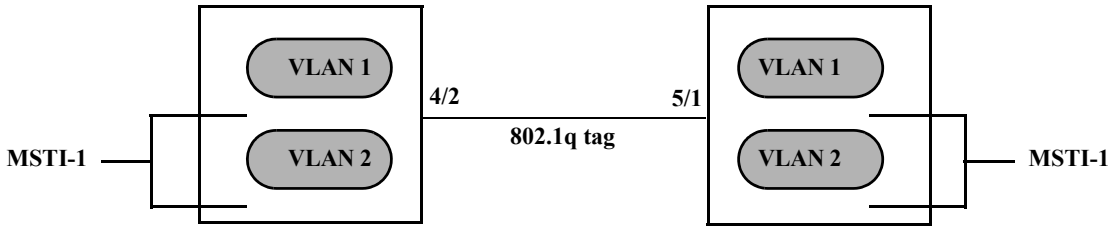


Figure 6-9 : Automatic VLAN Containment - AVC not enabled

In the above diagram, port 4/2 is the Root port and port 5/1 is a Designated port for MSTI 1. AVC is not enabled. If another link with the same speed and lower port numbers is added to default VLAN 1 on both switches, the new link becomes the root for MSTI 1 and the tagged link between VLAN 2 is blocked, as shown below:

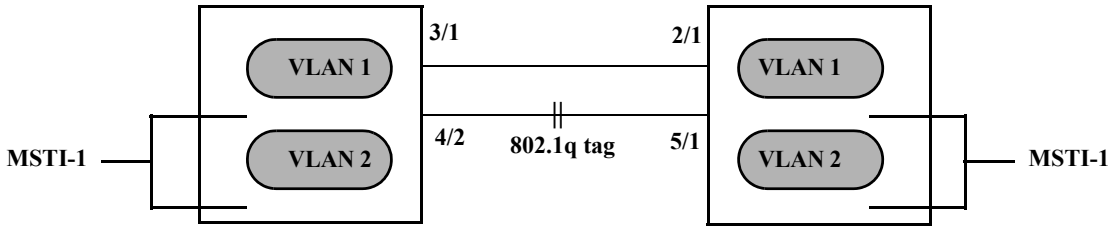


Figure 6-10 : Automatic VLAN Containment - AVC enabled

If AVC was enabled in the above example, AVC would have assigned the new link an infinite path cost value that would make this link undesirable as the root for MSTI 1.

Balancing VLANs across links according to their Multiple Spanning Tree Instance (MSTI) grouping is highly recommended to ensure that there is not a loss of connectivity during any possible topology changes. Enabling AVC on the switch is another way to prevent undesirable ports from becoming the root for an MSTI.

To change the default status of the AVC on the switch and to globally enable this feature for all MSTIs, use the **spantree auto-vlan-containment** command. Once AVC is globally enabled, then it is possible to disable AVC for individual MSTIs using the same command. For example, the following commands globally enable AVC and then disable it for MSTI 10:

```
-> spantree auto-vlan-containment enable
-> spantree msti 10 auto-vlan-containment disable
```

Note. An administratively set port path cost takes precedence and prevents AVC configuration of the path cost. The exception to this is if the port path cost is administratively set to zero, which resets the path cost to the default value. In addition, AVC does not have any effect on root bridges.

Configuring STP Port Parameters

The following sections provide information and procedures for using CLI commands to configure STP port parameters. These parameters determine the behavior of a port for a specific Spanning Tree instance.

When a switch is running in the per-VLAN STP mode, each VLAN is in essence a virtual STP bridge with its own STP instance and configurable parameters. To change STP port parameters while running in this mode, a VLAN ID is specified to identify the VLAN STP instance associated with the specified port. When a switch is running in the flat Spanning Tree mode, VLAN 1 is specified for the VLAN ID.

Only bridged ports participate in the Spanning Tree Algorithm. A port is considered bridged if it meets all the following criteria:

- Port is either a fixed (non-mobile) port, an 802.1Q tagged port, or a link aggregate logical port.
- Spanning tree is enabled on the port.
- Port is assigned to a VLAN that has Spanning Tree enabled.
- Port state (forwarding or blocking) is dynamically determined by the Spanning Tree Algorithm, not manually set.

The following is a summary of Spanning Tree port configuration commands. For more information about these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide.*

Commands	Used for ...
spantree cist	Configuring the port Spanning Tree status for the single flat mode instance.
spantree vlan	Configuring the port Spanning Tree status for a VLAN instance.
spantree priority	Configuring the priority value for the flat mode CIST instance, a Multiple Spanning Tree Instance (MSTI), or a per-VLAN mode VLAN instance.
spantree loop-guard	Enables or disables the STP loop-guard on a port or link aggregate.
spantree cist path-cost	Configuring the port path cost value for the single flat mode instance.
spantree msti path-cost	Configuring the port path cost value for a Multiple Spanning Tree Instance (MSTI).
spantree vlan path-cost	Configuring the port path cost value for a VLAN instance.
spantree cist mode	Configuring the port Spanning Tree mode (dynamic or manual) for the single flat mode instance.
spantree loop-guard	Configuring the port Spanning Tree mode (dynamic or manual) for a VLAN instance.
spantree cist connection	Configuring the port connection type for the single flat mode instance.
spantree vlan connection	Configuring the port connection type for a VLAN instance.
spantree cist admin-edge	Configures the connection type for a port or an aggregate of ports for the flat mode Common and Internal Spanning Tree (CIST).
spantree vlan admin-edge	Configures the connection type for a port or an aggregate of ports for a per-VLAN mode VLAN instance.

Commands	Used for ...
spantree cist auto-edge	Configures a port or an aggregate of ports for the flat mode Common and Internal Spanning Tree (CIST) as an edge port, automatically.
spantree vlan auto-edge	Configures a port or an aggregate of ports for the per-VLAN mode VLAN instance as an edge port, automatically.
spantree cist restricted-role	Configures the restricted role status for a port or an aggregate of ports for the flat mode Common and Internal Spanning Tree (CIST) as a restricted role port.
spantree vlan restricted-role	Configures a port or an aggregate of ports for the per-VLAN mode VLAN instance as a restricted role port.
spantree cist restricted-tcn	Configures a port or an aggregate of ports for the flat mode Common and Internal Spanning Tree (CIST) to support the restricted TCN capability.
spantree vlan restricted-tcn	Configures a port or an aggregate of ports for the per-VLAN mode VLAN instance to support the restricted TCN capability.
spantree cist txholdcount	Limits the transmission of BPDU through a given port for the flat mode Common and Internal Spanning Tree (CIST).
spantree vlan txholdcount	Limits the transmission of BPDU through a given port for the per-VLAN mode VLAN instance.
spantree pvst+compatibility	Configures the type of BPDU to be used on a port when PVST+ mode is enabled.

The following sections provide information and procedures for using Spanning Tree port configuration commands and also includes command examples.

Enabling/Disabling Spanning Tree on a Port

Spanning Tree is automatically enabled on all eligible ports. When Spanning Tree is disabled on a port, the port is put in a forwarding state for the specified instance. For example, if a port is associated with both VLAN 10 and VLAN 20 and Spanning Tree is disabled on the port for VLAN 20, the port state is set to forwarding for VLAN 20. However, the VLAN 10 instance still controls the port state as it relates to VLAN 10. This example assumes the switch is running in the per-VLAN Spanning Tree mode.

If the switch is running in the flat Spanning Tree mode, then disabling the port Spanning Tree status applies across all VLANs associated with the port. The flat mode instance is specified as the instance associated with the port, even if the port is associated with multiple VLANs.

To change the port Spanning Tree status for a VLAN instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree vlan** command. For example, the following commands enable Spanning Tree on port 8/1 for VLAN 10 and disable STP on port 6/2 for VLAN 20:

```
-> spantree vlan 10 port 8/1 enable
-> spantree vlan 20 port 6/2 disable
```

To change the port Spanning Tree status for the flat mode instance, use the **spantree cist** command. Note that this command is available when the switch is running in either mode (per-VLAN or flat). For example, the following command disables the Spanning Tree status on port 1/24 for the flat mode instance:

```
-> spantree cist port 1/24 disable
```

Spanning Tree on Link Aggregate Ports

Physical ports that belong to a link aggregate do not participate in the Spanning Tree Algorithm. Instead, the algorithm is applied to the aggregate logical link (virtual port) that represents a collection of physical ports.

To enable or disable the Spanning Tree status for a link aggregate, use the **spantree vlan** or **spantree cist** commands described above but specify a link aggregate control (ID) number instead of a slot and port. For example, the following command disables Spanning Tree for the link aggregate 10 association with VLAN 755:

```
-> spantree vlan 755 linkagg 10 disable
```

For more information about configuring an aggregate of ports, see Chapter 9, "Configuring Static Link Aggregation," and Chapter 10, "Configuring Dynamic Link Aggregation."

Enabling/Disabling Loop-guard

By default, loop-guard is disabled on ports associated with VLANs that have Spanning Tree enabled. This feature, when enabled prevents inconsistencies that cause network loops.

Use the **spantree loop-guard** command to enable or disable loop-guard on a port or link aggregate. For example, the following commands enable and disable loop-guard on port 1/2 of chassis 1:

```
-> spantree port 1/1/2 loop-guard enable
-> spantree port 1/1/2 loop-guard disable
```

To enable or disable loop-guard on a link aggregate:

```
-> spantree linkagg 1 loop-guard enable
-> spantree linkagg 1 loop-guard disable
```

Note. Use the **show spantree** and related commands to view the loop-guard related information for per-port, per-VLAN, CIST or MSTI instances.

Configuring Port Priority

A bridge port is identified within the Spanning Tree by its Port ID (a 16-bit or 32-bit hex number). The first 4 bits of the Port ID contain a priority value and the remaining 12 bits contain the physical switch port number. The port priority is used to determine which port offers the best path to the root when multiple paths have the same path cost. The port with the highest priority (lowest numerical priority value) is selected and the others are put into a blocking state. If the priority values are the same for all ports in the path, then the port with the lowest physical switch port number is selected.

Spanning Tree is automatically enabled on a port and the default port priority value is set. If the switch is running in the per-VLAN Spanning Tree mode, then the port priority applies to the specified VLAN

instance associated with the port. If the switch is running in the flat Spanning Tree mode, then the port priority applies across all VLANs associated with the port. The flat mode instance is specified as the port instance, even if the port is associated with multiple VLANs.

To change the port priority value for a VLAN regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree priority** command with the **vlan** and **port** parameters. For example, the following command sets the priority value as 3 for the port 10/1 association with VLAN ID 10:

```
-> spantree vlan 10 port 10/1 priority 3
```

To change the port priority value for the flat mode instance, use the **spantree priority** command with the **cist** and **port** parameters. Note that this command is available when the switch is running in either per-VLAN or flat mode. An instance number is not required. For example, the following command changes the priority value for port 1/24 for the flat mode instance to 15:

```
-> spantree cist port 1/24 priority 15
```

The port priority value is also configurable for a Multiple Spanning Tree Instance (MSTI). To configure this value for an MSTI, use the **spantree priority** command with the **msti** and **port** parameters. For example, the following command configures the priority value for port 1/12 for MSTI 10 to 5:

```
-> spantree msti 10 port 1/12 priority 5
```

Note that configuring the port priority value for a MSTI is allowed in both modes (per-VLAN and flat) only when the Spanning Tree protocol is set to MSTP.

Port Priority on Link Aggregate Ports

Physical ports that belong to a link aggregate do not participate in the Spanning Tree Algorithm. Instead, the algorithm is applied to the aggregate logical link (virtual port) that represents a collection of physical ports.

To change the priority for a link aggregate, use the **spantree priority** command with the **cist**, **msti**, or **vlan** parameters, as described above but specify a link aggregate control number instead of a slot and port number. For example, the following command sets the priority for the link aggregate 10 association with VLAN 755 to 9:

```
-> spantree vlan 755 linkagg 10 priority 9
```

For more information about configuring an aggregate of ports, see Chapter 9, “Configuring Static Link Aggregation,” and Chapter 10, “Configuring Dynamic Link Aggregation.”

Configuring Port Path Cost

The path cost value specifies the contribution of a port to the path cost towards the root bridge that includes the port. The root path cost is the sum of all path costs along this same path and is the value advertised in Configuration BPDU transmitted from active Spanning Tree ports. The lower the cost value, the closer the switch is to the root.

The type of path cost value used depends on which path cost mode is active (automatic or 32-bit). If the path cost mode is set to automatic, a 16-bit value is used when STP or RSTP is the active protocol and a 32-bit value is used when MSTP is the active protocol. If the mode is set to 32-bit, then a 32-bit path cost value is used regardless of which protocol is active. See “Configuring the Path Cost Mode” on page 6-31 for more information.

If a 32-bit path cost value is in use and the *path_cost* is set to zero, the following IEEE 802.1t recommended default path cost values based on link speed are used:

Link Speed	32-bit Path Cost Physical Port	32-bit Path Cost Link Aggregate (2/4/8/16 Port)
10 Mbps	2000000	1200000, 800000, 600000, 400000
100 Mbps	200000	120000, 80000, 60000, 40000
1G	20000	18000, 16000, 14000, 12000
2.5G	8000	7600, 7200, 6800, 6400
5G	4000	3800, 3600, 3400, 3200
10G	2000	1900, 1800, 1700, 1600
25G	800	780, 760, 740, 720
40G	500	480, 460, 440, 420
50G	400	380, 360, 340, 320
100G	200	180, 160, 140, 120

Is a 16-bit path cost value is in use and the *path_cost* is set to zero, the following IEEE 802.1D recommended default path cost values based on link speed are used:

Link Speed	16-bit Path Cost Physical Port	16-bit Path Cost Link Aggregate (2/4/8/16 Port)
10 Mbps	100	60, 40, 30, 20
100 Mbps	19	12, 9, 7, 5
1G	4	3
2.5G	4	3
5G	3	2
10G	2	1
25G	1	1
40G	1	1
50G	1	1
100G	1	1

Spanning Tree is automatically enabled on a port and the path cost is set to the default value. If the switch is running in the per-VLAN Spanning Tree mode, then the port path cost applies to the specified VLAN instance associated with the port. If the switch is running in the flat Spanning Tree mode, then the port path cost applies across all VLANs associated with the port. The flat mode instance is specified as the port instance, even if the port is associated with other VLANs.

The **spantree vlan path-cost** command configures the port path cost value for a VLAN instance when the switch is running in either mode (per-VLAN or flat). For example, the following command configures a 16-bit path cost value for port 8/1 for VLAN 10 to 19 (the port speed is 100 MB, 19 is the recommended value):

```
-> spantree vlan 10 port 8/1 path-cost 19
```

To change the port path cost value for the flat mode instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree cist path-cost** command. For example, the following command configures a 32-bit path cost value for port 1/24 for the flat mode instance to 20,000 (the port speed is 1 GB, 20,000 is the recommended value):

```
-> spantree cist port 1/24 path-cost 20000
```

The port path cost value is also configurable for a Multiple Spanning Tree Instance (MSTI). To configure this value for an MSTI, use the **spantree msti path-cost** command and specify the MSTI ID for the instance number. For example, the following command configures the path cost value for port 1/12 for MSTI 10 to 19:

```
-> spantree msti 10 port 1/12 path-cost 19
```

Note that configuring the port path cost value for a MSTI is allowed in both modes (per-VLAN and flat) only when the Spanning Tree protocol is set to MSTP.

Path Cost for Link Aggregate Ports

Physical ports that belong to a link aggregate do not participate in the Spanning Tree Algorithm. Instead, the algorithm is applied to the aggregate logical link (virtual port) that represents a collection of physical ports. Spanning Tree is automatically enabled on the aggregate logical link and the path cost value is set to the default value using the tables above.

To change the path cost value for a link aggregate, use the **spantree cist path cost**, **spantree msti path cost**, or **spantree vlan path cost** command with the **linkagg** parameter and a link aggregate control (ID) number. For example, the following command sets the path cost for link aggregate 10 associated with VLAN 755 to 19:

```
-> spantree vlan 755 linkagg 10 path-cost 19
```

For more information about configuring an aggregate of ports, see Chapter 9, “Configuring Static Link Aggregation,” and Chapter 10, “Configuring Dynamic Link Aggregation.”

Configuring Port Mode

There are two port modes supported: manual and dynamic. Manual mode indicates that the port was set by the user to a forwarding or blocking state. The port operates in the state selected until the state is manually changed again or the port mode is changed to dynamic. Ports operating in a manual mode state do not participate in the Spanning Tree Algorithm. Dynamic mode indicates that the active Spanning Tree Algorithm determines port state.

Spanning Tree is automatically enabled on the port and the port operates in the default mode. If the switch is running in the per-VLAN Spanning Tree mode, then the port mode applies to the specified VLAN instance associated with the port. If the switch is running in the flat Spanning Tree mode, then the port mode applies across all VLANs associated with the port. The flat mode instance is specified as the port instance, even if the port is associated with other VLANs.

To change the port Spanning Tree mode for a VLAN instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree loop-guard** command. For example, the following command sets the mode for port 8/1 for VLAN 10 to forwarding.

```
-> spantree vlan 10 port 8/1 mode forwarding
```

To change the port Spanning Tree mode for the flat mode instance, use the **spantree cist mode** command. Note that the command is available when the switch is running in either mode (per-VLAN or flat) and an

instance number is not required. For example, the following command configures the Spanning Tree mode on port 1/24 for the flat mode instance:

```
-> spantree cist port 1/24 mode blocking
```

Mode for Link Aggregate Ports

Physical ports that belong to a link aggregate do not participate in the Spanning Tree Algorithm. Instead, the algorithm is applied to the aggregate logical link (virtual port) that represents a collection of physical ports.

To change the port mode for a link aggregate, use the **spantree vlan mode** or the **spantree cist mode** command described above, but specify a link aggregate control (ID) number instead of a slot and port. For example, the following command sets the port mode for link aggregate 10 associated with VLAN 755 to blocking:

```
-> spantree vlan 755 linkagg 10 mode blocking
```

For more information about configuring an aggregate of ports, see Chapter 9, “Configuring Static Link Aggregation,” and Chapter 10, “Configuring Dynamic Link Aggregation.”

Configuring Port Connection Type

Specifying a port connection type is done when using the Rapid Spanning Tree Algorithm and Protocol (RSTP), as defined in the IEEE 802.1w standard. RSTP transitions a port from a blocking state directly to forwarding, bypassing the listening and learning states, to provide a rapid reconfiguration of the Spanning Tree in the event of a path or root bridge failure. Rapid transition of a port state depends on the configurable connection type of the port. These types are defined as follows:

- Point-to-point LAN segment (port connects directly to another switch).
- No point-to-point shared media LAN segment (port connects to multiple switches).
- Edge port (port is at the edge of a bridged LAN, does not receive BPDU and has only one MAC address learned). Edge ports, however, will operationally revert to a point to point or a no point to point connection type if a BPDU is received on the port.

A port is considered connected to a point-to-point LAN segment if the port belongs to a link aggregate of ports, or if auto negotiation determines if the port must run in full duplex mode, or if full duplex mode was administratively set. Otherwise, that port is considered connected to a no point-to-point LAN segment.

Rapid transition of a designated port to forwarding can only occur if the port connection type is defined as a point to point or an edge port. Defining a port connection type as a point to point or as an edge port makes the port eligible for rapid transition, regardless of what actually connects to the port. However, an alternate port is always allowed to transition to the role of root port regardless of the alternate port connection type.

Note. Configure ports that will connect to a host (PC, workstation, server, etc.) as edge ports so that these ports will transition directly to a forwarding state and not trigger an unwanted topology change when a device is connected to the port. If a port is configured as a point to point or no point to point connection type, the switch will assume a topology change when this port goes active and will flush and relearn all learned MAC addresses for the port's assigned VLAN.

If the switch is running in the per-VLAN Spanning Tree mode, then the connection type applies to the specified VLAN instance associated with the port. If the switch is running in the flat Spanning Tree mode, then the connection type applies across all VLANs associated with the port. The flat mode instance is referenced as the port instance, even if the port is associated with other VLANs.

By default, Spanning Tree is automatically enabled on the port, the connection type is set to auto point-to-point, and auto edge port detection is enabled. The auto point-to-point setting determines the connection type based on the operational status of the port. The auto edge port setting determines the operational edge port status for the port.

The **spantree vlan connection** and **spantree cist connection** commands are used to configure the port connection type for a VLAN instance or the CIST instance. See "Configuring the Edge Port Status" on page 6-41 for information about configuring the auto edge port status for a port.

To change the port connection type for a VLAN instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree vlan connection** command. For example, the following command defines the connection type for port 8/1 associated with VLAN 10.

```
-> spantree vlan 10 port 8/1 connection autoptp
```

To change the port Spanning Tree mode for the flat mode instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree cist connection** command. For example, the following command configures the connection type for port 1/24 for the flat mode instance:

```
-> spantree cist port 1/24 connection ptp
```

Note. The **spantree vlan connection** and **spantree cist connection** commands only configure one port at a time.

Connection Type on Link Aggregate Ports

Physical ports that belong to a link aggregate do not participate in the Spanning Tree Algorithm. Instead, the algorithm is applied to the aggregate logical link (virtual port) that represents a collection of physical ports. To change the port connection type for a link aggregate, use the **spantree vlan connection** or **the spantree cist connection** command described above, but specify a link aggregate control (ID) number instead of a slot and port. For example, the following command defines the connection type for the link aggregate 10 association with VLAN 755:

```
-> spantree vlan 755 linkagg 10 connection autoptp
```

For more information about configuring an aggregate of ports, see Chapter 9, "Configuring Static Link Aggregation," and Chapter 10, "Configuring Dynamic Link Aggregation."

Configuring the Edge Port Status

There are two methods for determining the edge port status for a port or link aggregate:

- Configuring the automatic edge (auto edge) port status. The status (enabled or disabled) of this Spanning Tree port parameter specifies whether or not the Spanning Tree automatically determines the operational edge port status for a port. This method is enabled by default.
- Configuring the administrative edge (admin edge) port status. The status (enabled or disabled) of this Spanning Tree port parameter is used to determine the edge port status when the auto edge port status is disabled. This method is disabled by default.

To configure the edge port status for the flat mode instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree cist auto-edge** command or the **spantree cist admin-edge** command. For example:

```
-> spantree cist port 8/23 auto-edge enable
-> spantree cist port 8/23 admin-edge disable
```

To configure the edge port status for a VLAN instance regardless of which mode (per-VLAN or flat) is active for the switch, use the **spantree vlan auto-edge** command or the **spantree vlan admin-edge** command. For example:

```
-> spantree vlan 10 port 8/23 auto-edge enable
-> spantree vlan 10 port 8/23 admin-edge disable
```

Note. If **auto-edge** is enabled on a port, then the **admin-edge** value is overridden.

Restricting Port Roles (Root Guard)

All ports are automatically eligible for root port selection. A port in a CIST/MSTI instance or per-VLAN instance can be prevented from becoming the root port by restricting the role of the port (also referred to as enabling root guard). This is done using the **spantree cist restricted-role** command or the **spantree vlan restricted-role** command regardless of which mode (per-VLAN or flat) is active for the switch. For example:

```
-> spantree cist port 1/2 restricted-role enable
-> spantree cist linkagg 10 restricted-role enable
-> spantree vlan 100 port 8/1 restricted-role enable
-> spantree vlan 20 linkagg 1 restricted-role enable
```

Note that the above commands also provide optional syntax; **restricted-role** or **root-guard**. For example, the following two commands perform the same function:

```
-> spantree vlan port 2/1 restricted-role enable
-> spantree vlan port 2/1 root-guard enable
```

When root guard is enabled for a port, it cannot become the root port, even if it is the most likely candidate for becoming the root port. However, this same port is designated as the alternate port when the root port is selected.

Enabling the restricted role status is used by network administrators to prevent bridges external to the core region of the network from influencing the Spanning Tree topology. However, note that enabling the restricted role status for a port may impact connectivity within the network.

Restricting TCN Propagation

All ports automatically propagate Topology Change Notifications (TCN) or Topology Changes (TC) to other ports. To restrict a port from propagating topology changes and notifications, use the **spantree cist restricted-tcn** command or the **spantree vlan restricted-tcn** command regardless of which mode (per-VLAN or flat) is active for the switch. For example:

```
-> spantree cist port 2/2 restricted-tcn enable
-> spantree cist linkagg 5 restricted-tcn enable
-> spantree vlan 10 port 1/5 restricted-tcn enable
-> spantree vlan 20 linkagg 1 restricted-tcn enable
```

Enabling the restricted TCN status is used by network administrators to prevent bridges external to the core region of the network from causing unnecessary MAC address flushing in that region. However, note that enabling the restricted TCN status for a port may impact Spanning Tree connectivity.

Limiting BPDU Transmission

The number of BPDUs to be transmitted per port per second can be limited using the **spantree cist txholdcount** command for a CIST instance or the **spantree vlan txholdcount** command for a per-VLAN instance. Both of these commands apply to all ports and link aggregates and are supported when the switch is running in either the per-VLAN mode or the flat mode. For example:

```
-> spantree cist txholdcount 5
-> spantree vlan 10 txholdcount 5
```

Sample Spanning Tree Configuration

This section provides an example network configuration in which the Spanning Tree Algorithm and Protocol has calculated a loop-free topology. In addition, a tutorial is also included that provides steps on how to configure the example network topology using the Command Line Interface (CLI).

Note that the following example network configuration illustrates using switches operating in the per-VLAN Spanning Tree mode and using RSTP (802.1w) to calculate a single data path between VLANs. See "MST General Overview" on page 6-12 for an overview and examples of using MSTP (802.1s).

Example Network Overview

The following diagram shows a four-switch network configuration with an active Spanning Tree topology, which was calculated based on both configured and default Spanning Tree parameter values:

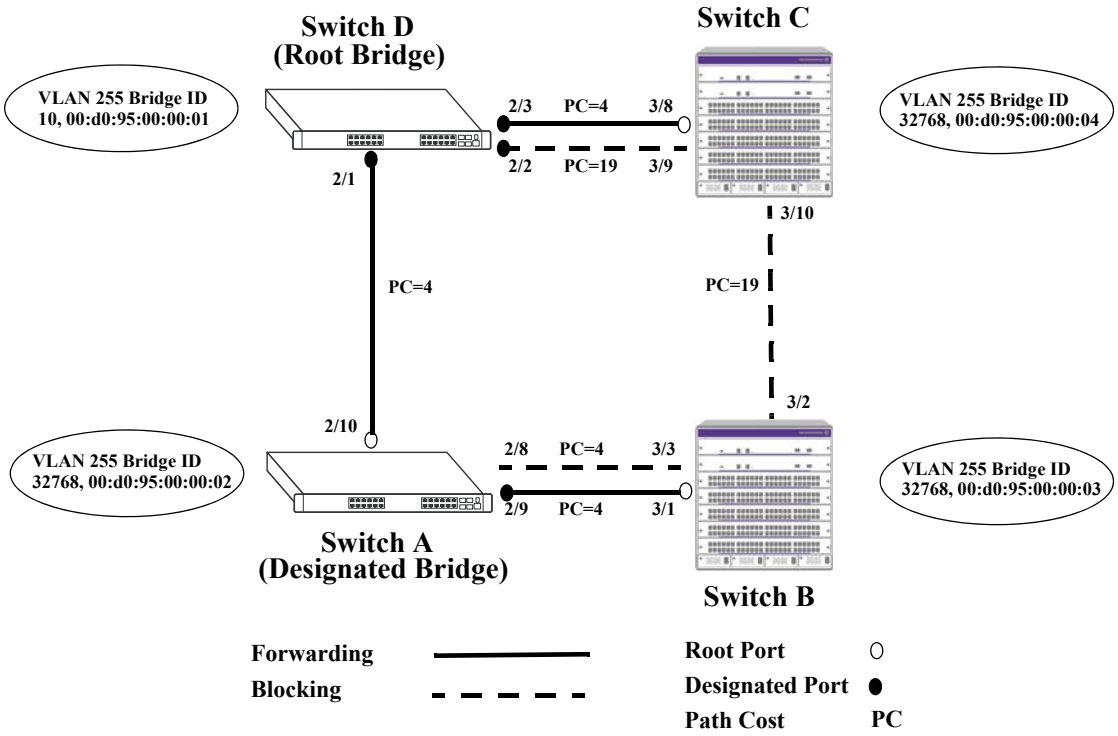


Figure 6-11 : Example Active Spanning Tree Topology

In the above example topology:

- Each switch is operating in the per-VLAN Spanning Tree mode by default.
- Each switch configuration has a VLAN 255 defined. The Spanning Tree administrative status for this VLAN was enabled by default when the VLAN was created.
- VLAN 255 on each switch is configured to use the 802.1w (rapid reconfiguration) Spanning Tree Algorithm and Protocol.
- Ports 2/1-3, 2/8-10, 3/1-3, and 3/8-10 provide connections to other switches and are all assigned to VLAN 255 on their respective switches. The Spanning Tree administrative status for each port is enabled by default.

- The path cost for each port connection defaults to a value based on the link speed. For example, the connection between Switch B and Switch C is a 100 Mbps link, which defaults to a path cost of 19.
- VLAN 255 on Switch D is configured with a Bridge ID priority value of 10, which is less than the same value for VLAN 255 configured on the other switches. As a result, VLAN 255 was elected the Spanning Tree root bridge for the VLAN 255 broadcast domain.
- A root port is identified for VLAN 255 on each switch, except the root VLAN 255 switch. The root port identifies the port that provides the best path to the root VLAN.
- VLAN 255 on Switch A was elected the designated bridge because it offers the best path cost for Switch B to the root VLAN 255 on Switch D.
- Port 2/9 on Switch A is the designated port for the Switch A to Switch B connection because Switch A is the designated bridge for Switch B.
- Redundant connections exist between Switch D and Switch C. Ports 2/2 and 3/9 are in a discarding (blocking) state because this connection has a higher path cost than the connection provided through ports 2/3 and 3/8. As a result, a network loop condition is avoided.
- Redundant connections also exist between Switch A and Switch B. Although the path cost value for both of these connections is the same, ports 2/8 and 3/3 are in a discarding state because their port priority values (not shown) are higher than the same values for ports 2/10 and 3/1.
- The ports that provide the connection between Switch B and Switch C are in a discarding (blocking) state, because this connection has a higher path cost than the other connections leading to the root VLAN 255 on Switch D. As a result, a network loop is avoided.

Example Network Configuration Steps

The following steps provide a quick tutorial that configures the active Spanning Tree network topology shown in the diagram on page 6-43.

1 Create VLAN 255 on Switches A, B, C, and D with “Marketing IP Network” for the VLAN description on each switch using the following command:

```
-> vlan 255 name "Marketing IP Network"
```

2 Assign the switch ports that provide connections between each switch to VLAN 255. For example, the following commands entered on Switches A, B, C, and D, respectively, assign the ports shown in the example network diagram on page 6-43 to VLAN 255:

```
-> vlan 255 members port 2/8-10 untagged
-> vlan 255 members port 3/1-3 untagged
-> vlan 255 members port 3/8-10 untagged
-> vlan 255 members port 2/1-3 untagged
```

3 Change the Spanning Tree protocol for VLAN 255 to RSTP (Rapid Spanning Tree Protocol) on each switch using the following command:

```
-> spantree vlan 255 protocol rstp
```

4 Change the bridge priority value for VLAN 255 on Switch D to **10** using the following command (leave the priority for VLAN 255 on the other three switches set to the default value):

```
-> spantree vlan 255 priority 10
```

VLAN 255 on Switch D has the lowest Bridge ID priority value of all four switches, which qualifies it as the Spanning Tree root VLAN for the VLAN 255 broadcast domain.

Note. To verify the VLAN 255 Spanning Tree configuration on each switch use the following **show** commands. The following outputs are for example purposes only and not match values shown in the sample network configuration:

```
-> show spantree vlan 255
Spanning Tree Parameters for Vlan 255
  Spanning Tree Status :                    ON,
  Protocol             :       IEEE Rapid STP,
  mode                 : Per VLAN (1 STP per Vlan),
  Priority             :       32768 (0x8000),
  Bridge ID            :   8000-00:e0:b1:e7:09:a3,
  Designated Root      :   8000-00:e0:b1:e7:09:a3,
  Cost to Root Bridge  :                     8,
  Root Port            :               1/1/48,
  TxHoldCount          :                     3,
  Topology Changes     :                   101,
  Topology age         :             01:05:30,
  Topology Change Port :               1/1/48,
    Current Parameters (seconds)
      Max Age              =    20,
      Forward Delay        =    15,
      Hello Time           =     2
    Parameters system uses when attempting to become root
      System Max Age       =    20,
      System Forward Delay =    15,
      System Hello Time    =     2

-> show spantree vlan 255 ports
Spanning Tree Port Summary for Vlan 1
        Oper  Path   Desig      Prim. Op  Op  Loop
Port     St   Cost    Cost Role Port  Cnx Edg Guard Desig Bridge ID        Note
-------+----+-------+-----+----+-----+---+---+-----+----------------------+-----
  1/1/1 FORW     100     8 DESG 1/1/1 PTP  NO  DIS  8000-e8:e7:32:a4:63:21
  1/1/2  DIS       0     0  DIS 1/1/2  NS  NO  DIS  0000-00:00:00:00:00:00
  1/1/4  DIS       0     0  DIS 1/1/4  NS  NO  DIS  0000-00:00:00:00:00:00
  1/1/5  DIS       0     0  DIS 1/1/5  NS  NO  DIS  0000-00:00:00:00:00:00
  1/1/6  DIS       0     0  DIS 1/1/6  NS  NO  DIS  0000-00:00:00:00:00:00
  1/1/7  DIS       0     0  DIS 1/1/7  NS  NO  DIS  0000-00:00:00:00:00:00
  1/1/8  DIS       0     0  DIS 1/1/8  NS  NO  DIS  0000-00:00:00:00:00:00
  1/1/9  DIS       0     0  DIS 1/1/9  NS  NO  DIS  0000-00:00:00:00:00:00
 1/1/12  DIS       0     0  DIS 1/1/12 NS  NO  DIS  0000-00:00:00:00:00:00
```

Sample MST Region Configuration

An MST region identifies a group of MSTP switches that is seen as a single, flat mode instance by other regions and/or non-MSTP switches. A region is defined by three attributes: name, revision level, and a VLAN-to-MSTI mapping. Switches configured with the same value for all three of these attributes belong to the same MST region.

Note. An additional configurable MST region parameter defines the maximum number of hops authorized for the region but is not considered when determining regional membership.The maximum hops value is the value used by all bridges within the region when the bridge is acting as the root of the MST region.

This section provides a tutorial for defining a sample MST region configuration, as shown in the diagram below:

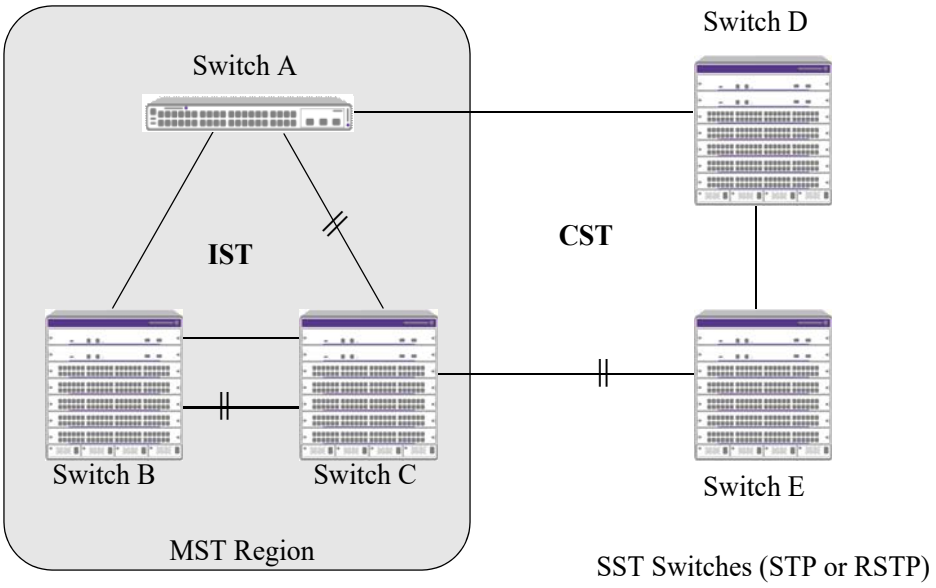


Figure 6-12 : Sample MST Region Configuration

In order for switches A, B, and C in the above diagram to belong to the same MST region, they must all share the same values for region name, revision level, and configuration digest (VLAN-to-MSTI mapping).

The following steps are performed on each switch to define **ALE Marketing** as the MST region name, **2000** as the MST region revision level, map exiting VLANs to existing MSTIs, and **3** as the maximum hops value for the region:

1 Configure an MST Region name using the **spantree mst region name** command. For example:

```
-> spantree mst region name "ALE Marketing"
```

2 Configure the MST Region revision level using the **spantree mst region revision-level** command. For example:

```
-> spantree mst region revision-level 2000
```

3 Map VLANs 100 and 200 to MSTI 2 and VLANs 300 and 400 to MSTI 4 using the **spantree msti vlan** command to define the configuration digest. For example:

```
-> spantree msti 2 vlan 100 200
-> spantree msti 4 vlan 300 400
```

See the "Sample MSTI Configuration" on page 6-48 for a tutorial on how to create and map MSTIs to VLANs.

4 Configure **3** as the maximum number of hops for the region using the **spantree mst region max-hops** command. For example:

```
-> spantree mst region max-hops 3
```

Note. (*Optional*) Verify the MST region configuration on each switch with the **show spantree mst** command. For example:

```
-> show spantree mst region
Configuration Name       = ALE Marketing,
Revision Level           = 2000,
Configuration Digest     = 0x922fb3f 31752d68 67fe1155 d0ce8380,
Revision Max hops        = 3,
Cist Instance Number                                    = 0
```

All switches configured with the exact same values as shown in the above example are considered members of the ALE Marketing MST region.

Sample MSTI Configuration

By default, the Spanning Tree software is active on all switches and operating in the per-VLAN mode using 802.1w RSTP. A loop-free network topology is automatically calculated based on default 802.1w RSTP switch, bridge, and port parameter values.

Using Multiple Spanning Tree (MST) requires configuration changes to the default Spanning Tree values (mode and protocol) as well as defining specific MSTP parameters and instances.

The following steps provide a tutorial for setting up a sample MSTP configuration, as shown in the diagram below:

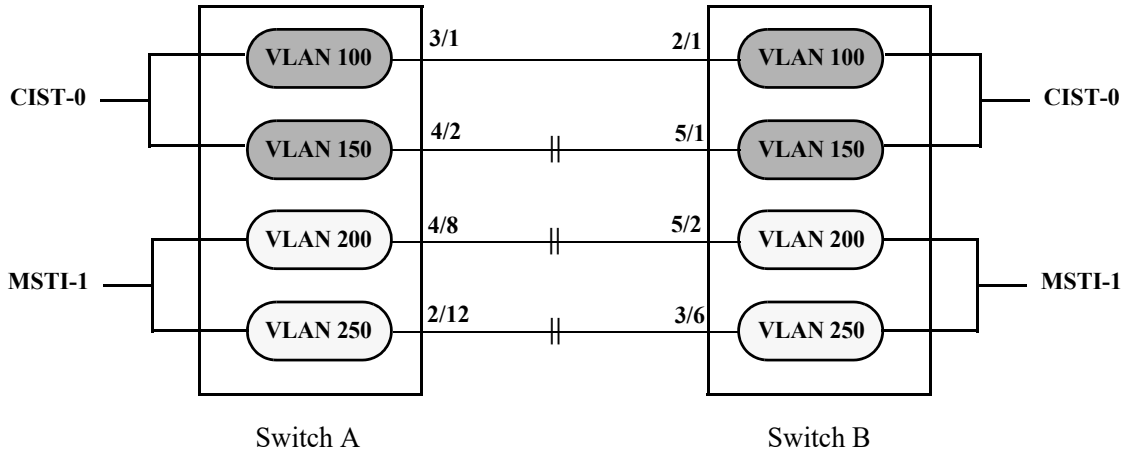


Figure 6-13 : Flat Mode MSTP Quick Steps Example

1 Change the Spanning Tree operating mode, if necessary, on Switch A and Switch B from per-VLAN to flat mode using the **spantree mode** command. For example:

```
-> spantree mode flat
```

Note that defining an MSTP configuration requires the use of explicit Spanning Tree commands, which are available in both the flat and per-VLAN mode. As a result, this step is optional. See "Using Spanning Tree Configuration Commands" on page 6-26 for more information.

2 Change the Spanning Tree protocol to MSTP using the **spantree protocol** command. For example:

```
-> spantree protocol mstp
```

3 Create VLANs 100, 200, 300, and 400 using the **vlan** command. For example:

```
-> vlan 100
-> vlan 150
-> vlan 200
-> vlan 250
```

4 Assign switch ports to VLANs, as shown in the above diagram, using the **vlan members untagged** command. For example, the following commands assign ports 3/1, 4/2, 4/8, and 2/12 to VLANs 100, 150, 200, and 250 on Switch A:

```
-> vlan 100 members port 3/1 untagged
-> vlan 150 members port 4/2 untagged
-> vlan 200 members port 4/8 untagged
-> vlan 250 members port 2/12 untagged
```

The following commands assign ports 2/1, 5/1, 5/2, and 3/6 to VLANs 100, 150, 200, and 250 on Switch B:

```
-> vlan 100 members port 2/1 untagged
-> vlan 150 members port 5/1 untagged
-> vlan 200 members port 5/2 untagged
-> vlan 250 members port 3/6 untagged
```

5 Create one MSTI using the **spantree msti** command. For example:

```
-> spantree msti 1
```

6 Assign VLANs 200 and 250 to MSTI 1. For example:

```
-> spantree msti 1 vlan 100 200
```

All VLANs are associated with the CIST instance. As a result, VLANs 100 and 150 do not require any configuration to map them to the CIST instance.

7 Configure the port path cost (PPC) for all ports on both switches associated with MSTI 1 to a PPC value that is lower than the PPC value for the ports associated with the CIST instance using the **spantree msti path-cost** command. For example, the PPC for ports associated with the CIST instance is set to the default of 200,000 for 100 MB connections. The following commands change the PPC value for ports associated with the MSTI 1 to 20,000:

```
-> spantree msti 1 port 4/8 path-cost 20000
-> spantree msti 1 port 2/12 path-cost 20000
-> spantree msti 1 port 5/2 path-cost 20000
-> spantree msti 1 port 3/6 path-cost 20000
```

Note. In this example, port connections between VLANs 150, 200, and 250 are blocked on each switch initially, as shown in the diagram on page 6-48. This is because in flat mode MSTP, each instance is active on all ports resulting in a comparison of connections independent of VLAN and MSTI associations.

To avoid this and allow VLAN traffic to flow over separate data paths based on MSTI association, Step 7 of this tutorial configures a superior port path cost value for ports associated with MSTI 1. As a result, MSTI 1 selects one of the data paths between its VLANs as the best path, rather than the CIST data paths, as shown in the diagram on page 6-50.

.

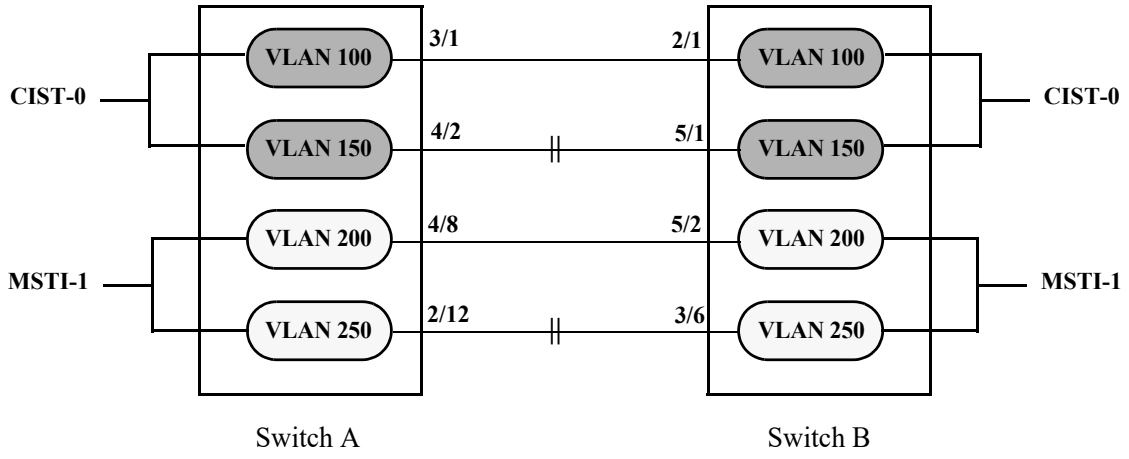


Figure 6-14 : Flat Mode MSTP with Superior MSTI 1 PPC Values

Note. Of the two data paths available to MSTI 1 VLANs, one is blocked because it is seen as redundant for that instance. In addition, the CIST data path remains available for CIST VLAN traffic.

Another solution to this scenario is to assign all VLANs to an MSTI, leaving no VLANs controlled by the CIST. As a result, the CIST BPDU contains only MSTI information. See "How MSTP Works" on page 6-12 for more information.

Verifying the Spanning Tree Configuration

To display information about the Spanning Tree configuration on the switch, use the show commands listed below:

show spantree cist	Displays the Spanning Tree bridge configuration for the flat mode Common and Internal Spanning Tree (CIST) instance.
show spantree msti	Displays Spanning Tree bridge information for a Multiple Spanning Tree Instance (MSTI).
show spantree vlan	Displays the Spanning Tree bridge information for a VLAN instance.
show spantree cist ports	Displays Spanning Tree port information for the flat mode Common and Internal Spanning Tree (CIST) instance.
show spantree msti ports	Displays Spanning Tree port information for a flat mode Multiple Spanning Tree Instance (MSTI).
show spantree vlan ports	Displays Spanning Tree port information for a VLAN instance.
show spantree mst	Displays the Multiple Spanning Tree (MST) information for a MST region or the specified port or link aggregate on the switch.
show spantree cist vlan-map	Displays the range of VLANs associated with the flat mode Common and Internal Spanning Tree (CIST) instance.
show spantree msti vlan-map	Displays the range of VLANs associated with the specified Multiple Spanning Tree Instance (MSTI).
show spantree map-msti	Displays the Multiple Spanning Tree Instance (MSTI) that is associated to the specified VLAN.
show spantree mode	Displays the current global Spanning Tree mode parameter values for the switch, such as the current running mode (per-VLAN or flat) for the switch

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. An example of the output for the **show spantree vlan** and **show spantree vlan ports** commands is also given in "Example Network Configuration Steps" on page 6-44.

7 Configuring Shortest Path Bridging

The OmniSwitch supports Shortest Path Bridging MAC (SPBM), as defined in the IEEE 802.1aq standard. SPBM uses the Provider Backbone Bridge (PBB) network model to encapsulate (using IEEE 802.1ah headers) and tunnel customer traffic through the network backbone. The shortest path trees upon which the PBB network infrastructure operates are determined using a version of the Intermediate System-to-Intermediate System (IS-IS) link state protocol that supports TLV extensions for SPB (ISIS-SPB).

Incorporating SPBM into the data center infrastructure provides the following benefits:

- Transparently extends Layer 2 connections (VLAN segments) across a large virtual service Layer 2 backbone network.
- Maintains a loop-free network while providing efficient use of available bandwidth, especially in a mesh topology. All connections between all switches in the topology remain active (no blocking of redundant links).
- A shortest path is automatically calculated between each bridge and every other bridge in the data center mesh, resulting in low latency and sub-second convergence times needed to support critical data center bridging requirements.
- Can process a large number of customer MAC addresses without overrunning provider network resources. Customer MAC addresses are only learned on Backbone Edge Bridges (BEB), where customer traffic is then encapsulated and tunneled through the network core infrastructure. Backbone Core Bridges (BCB) do not have to learn any customer MAC addresses.
- Provides a clear separation of customer traffic (between different customers and between the provider network domain). Entry points for customer traffic are clearly defined on the participating BEBs. Customer traffic is identified and associated with a specific service instance bound to the PBB infrastructure.
- Integration with Virtual Machine Network Profiles (vNPs) to support virtual machine (VM) discovery and mobility.

In This Chapter

This chapter provides an overview about how Shortest Path Bridging MAC (SPBM) works and how to configure SPBM through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

This chapter includes the following topics:

- "SPBM Parameter Defaults" on page 7-3.
- "SPBM Interface Defaults" on page 7-3.
- "SPBM Service Defaults" on page 7-4.
- "Shortest Path Bridging Overview" on page 7-5.
- "SPBM Pseudo-Wire (E-LINE Transparent) Service" on page 7-14.
- "Remote Fault Propagation for SPBM Services" on page 7-15.
- "IP over SPBM" on page 7-18.
- "SPB Over Shared Ethernet" on page 7-23.
- "SPB In-Band Management" on page 7-25.
- "Interaction With Other Features" on page 7-32.
- "Quick Steps for Configuring SPBM" on page 7-36.
- "Configuring SPBM" on page 7-39.
- "Configuring an SPB Pseudo-Wire Service" on page 7-63.
- "Configuring Remote Fault Propagation for SPBM" on page 7-66.
- "Configuring IP over SPB" on page 7-73.
- "Configuring SPB Over Shared Ethernet" on page 7-101.
- "Verifying the SPB Backbone and Services" on page 7-107.

SPBM Parameter Defaults

Parameter Description	Command	Default
ISIS-SPB status for the switch.	**spb isis admin-state**	Disabled
Equal Cost Tree (ECT) ID number for the backbone VLAN (BVLAN).	**spb isis bvlan ect-id**	1 or next available ECT ID number on the local switch.
Control BVLAN for the switch.	**spb isis control-bvlan**	None
The BVLAN tandem multicast mode (only applies to associated SPB services running in tandem mode).	**spb isis bvlan tandem-multicast-mode**	Source and Group (S, G)
Priority value for the ISIS-SPB instance.	**spb isis bridge-priority**	32768
Wait time intervals, in milliseconds, for shortest path first (SPF) calculations.	**spb isis spf-wait**	maximum wait: 1000 initial wait : 100 second wait : 300
Wait time intervals, in milliseconds, for link state PDU (LSP) transmissions.	**spb isis lsp-wait**	maximum wait: 1000 initial wait : 0 second wait : 300
Graceful restart status for the switch.	**spb isis graceful-restart**	Enabled
Graceful restart helper status for the switch.	**spb isis graceful-restart helper**	Enabled

SPBM Interface Defaults

Parameter Description	Command	Default
SPB interface status	**spb isis interface**	Disabled
SPB interface time interval between each hello packet transmission.	**spb isis interface hello-interval**	9 seconds
SPB interface hello multiplier used to determine hello packet hold time.	**spb isis interface hello-multiplier**	3
SPB interface link cost to reach the peer bridge.	**spb isis interface metric**	10
SPB interface type (P2P or multiple access).	**spb isis interface type**	Point-to-point (P2P)
SPB interface priority (applies only to multiple access interfaces)	**spb isis interface priority**	64

SPBM Service Defaults

By default, there are no SPBM service components configured for the switch. However, when a service is created, the following default values apply:

Parameter Description	Command	Default
SPB service administrative status.	**service admin-state**	Disabled
SPB service multicast replication mode.	**service multicast-mode**	Head-end
SPB service VLAN translation.	**service vlan-xlation**	Disabled
SPB service maximum transmission unit (MTU) value.	Not configurable at this time	9194
SPB service statistics collection.	**service stats**	Disabled
SPB service description.	**service description**	None.
Default profile automatically applied to access ports.	**service access l2profile**	def-access-profile
Layer 2 profile that specifies how control packets are processed on service access ports.	**service l2profile**	def-access-profile: STP, GVRP, MVRP = tunnel 802.3ad = peer 802.1x, 802.1ab, AMAP = drop CSCO PDU, VLAN, uplink = drop
VLAN translation for the service access port.	**service access vlan-xlation**	Disabled
Service access point (SAP) administrative status.	**service sap admin-state**	Enabled
SAP encapsulation.	**service sap**	0 (untagged traffic).
SAP trust mode.	**service sap trusted**	Trusted
SAP statistics collection.	**service sap stats**	Disabled
SAP description.	**service sap description**	None

Shortest Path Bridging Overview

The OmniSwitch implementation of Shortest Path Bridging (SPB) supports SPB MAC (SPBM) as defined in the IEEE 802.1aq standard. SPBM is defined for use in Provider Backbone Bridge (PBB) networks as specified in the IEEE 802.1ah standard.

SPBM provides a mechanism to automatically define a shortest path tree (SPT) bridging configuration through a Layer 2 Ethernet network. SPBM Ethernet services use this configuration to encapsulate and tunnel data through the PBB network. The following main components of the OmniSwitch implementation of SPBM provide this type of functionality:

- **ISIS-SPB—**A version of the Intermediate to Intermediate System (IS-IS) link state protocol that supports SPB TLV extensions. SPBM uses ISIS-SPB to build sets of symmetric shortest path trees (SPTs) between any SPB switch.
- **Provider Backbone Bridge (PBB) IEEE 802.1ah**— Defines a MAC-in-MAC data encapsulation path for PBB networks that is supported by SPBM.
- **Provider Backbone Bridge Network (PBBN)—**A network comprised of Backbone Edge Bridges (BEBs) and Backbone Core Bridges (BCB) that is used to interconnect Provider Bridge Networks (PBN) with other networks.
- **Backbone Edge Bridge (BEB)—**An SPB switch positioned at the edge of the PBB network that learns and encapsulates (adds an 802.1ah backbone header to) customer frames for transport across the backbone network. The BEB interconnects the customer network space with PBB network space.
- **Backbone Core Bridge (BCB)—**An SPB node that resides inside the PBB network core. The BCB employs the same BVLAN on two or more network ports. This BVLAN does not terminate on the switch itself; traffic ingressing on an SPB network port is switched to other SPB network ports. As a result, the BCB does not have to learn any of the customer MAC addresses. It mainly serves as a transit bridge for the PBB network.
- **SPBM Service—**An OmniSwitch Service Manager service configured on the BEBs. Each service maps to a service instance identifier (I-SID) which is bound to a backbone VLAN. One backbone VLAN can accommodate multiple I-SIDs.
- **Backbone VLAN (BVLAN)—**A VLAN that serves as a transport VLAN for the SPBM service instances and to connect SPB bridges together through SPT sets. Unlike standard VLANs, BVLANs do not learn source MAC addresses or flood unknown destination or multicast frames. Instead, BVLANs only forward on the basis of the forwarding database (FDB) as populated by the ISIS-SPB protocol.

The following diagram shows how SPBM uses the above components to tunnel customer traffic through a Provider Backbone Bridge Network:

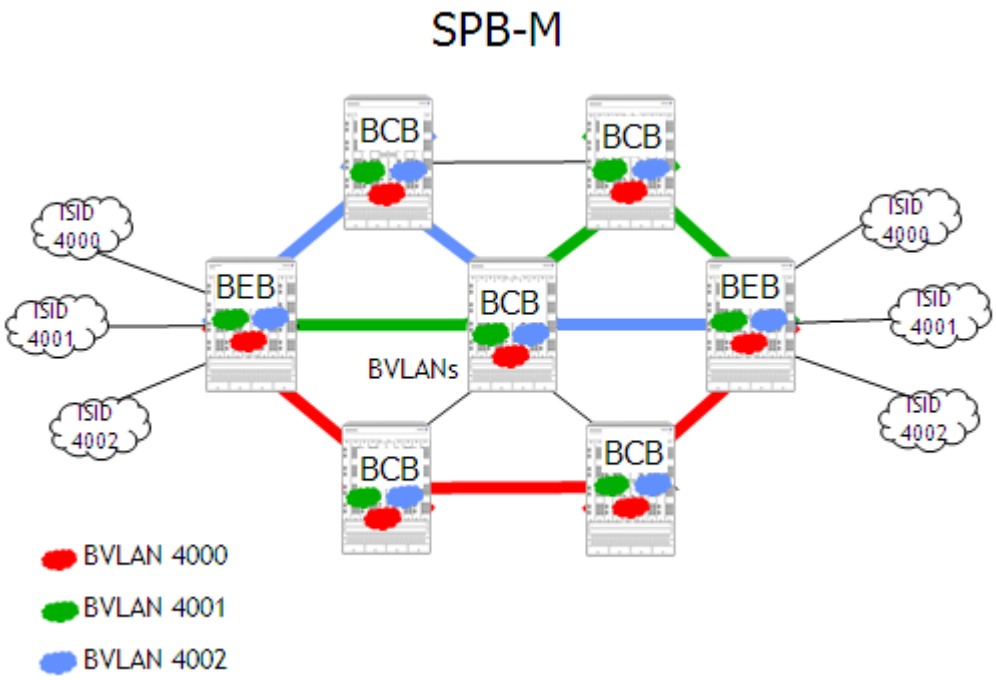


Figure 7-1 : SPBM Network Components

In this network,

- The BEBs are SPBM capable (ISIS-SPB configured and enabled) and form a shortest path bridging network that also includes the SPBM capable Backbone Core Bridges (BCBs).
- Each bridge calculates a shortest path tree (SPT) for each BVLAN with itself as the root of each tree.
- SPB Ethernet service instances identified by I-SIDs are created on each BEB. Each I-SID is associated with a BVLAN ID. The BVLAN is configured on each bridge (BEB and BCB) in the backbone network. However, the I-SID itself and the I-SID association with the BVLAN is only configured on each BEB that will service customer traffic.
- A Service Access Point (SAP) is configured on each BEB to identify the access port on which customer traffic will enter the PBBN, the SPB service instance that will tunnel the traffic through the network, and the type of customer traffic to forward (for example, only specific CVLAN IDs, untagged traffic only, or all tagged traffic). Basically, the SAP binds access ports and the specified customer traffic received on those ports to the service.
- Layer 2 traffic from the connected edge networks enters the BEBs through access ports. The SAP configuration on the receiving access port is applied to classify which frames are mapped to which services, if any.
- Classified traffic is then encapsulated into 802.1ah frames by the BEB before the frames are transmitted through the backbone network.
- The 802.1ah encapsulated frames are forwarded on the shortest path through the entire PBBN to reach the intended destination BEB. The BCBs switch traffic based on the destination backbone MAC address (BMAC)—bridge MAC address of the BEB—provided in the 802.1ah header and do not process any I-SID information in the frame.

SPBM Shortest Path Trees

The shortest path between two points is a straight line. Shortest Path Bridging (SPB) implements frame forwarding on the shortest path between any two bridges in an Ethernet network. The shortest path trees (SPTs) calculated by SPB provide the shortest and most efficient path to and from the intended destination. SPTs are formed along the direct, straight-line links between switches to make up an overall path through the topology that provides a robust, efficient direction for network traffic to travel.

The SPBM network topology consists of two layers:

- **The backbone infrastructure (control plane) layer**. ISIS-SPB builds the backbone layer by defining loop-free, shortest path trees (SPTs) through the backbone network.
- **The services (data plane) layer**. The service layer is based on the Provider Backbone Bridging (PBB) framework as defined in the IEEE 802.1ah standard. SPBM supports the 802.1ah MAC-in-MAC method for data encapsulation. SPBM services transport the encapsulated traffic over the ISIS-SPB infrastructure. (See "SPB Services" on page 7-11 for more information).

This section contains an example of ISIS-SPB operations in a small SPBM network. In addition to describing how shortest path trees are created in the BVLAN domain, the flow of unicast and multicast traffic through the network, this example also shows the benefits of using SPB over Spanning Tree for VLAN traffic distribution.

Spanning Tree

The following diagram shows an example Provider Backbone Bridge (PBB) network with a single backbone VLAN using the Spanning Tree protocol for network loop protection with the same path cost on all links:

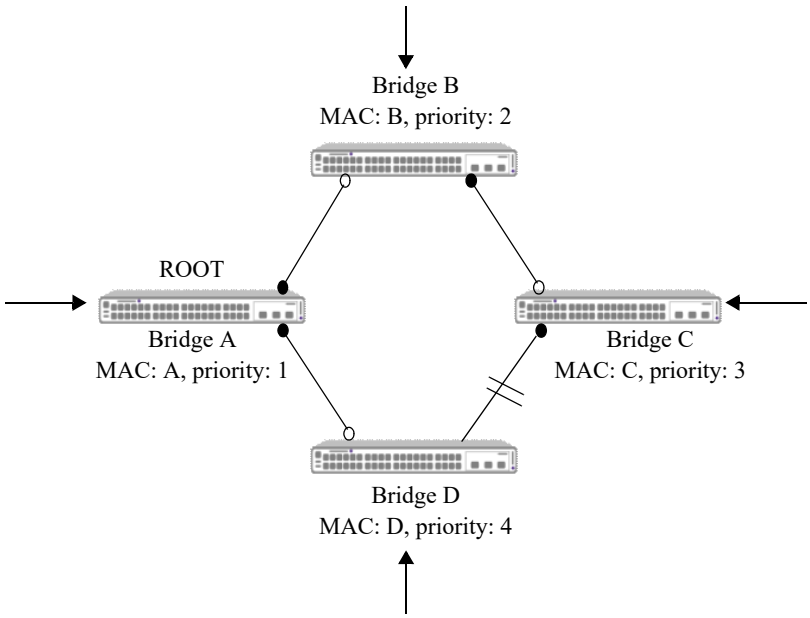


Figure 7-2 : Spanning Tree Topology

In this example, Bridge A is the Root bridge. As a result, customer traffic entering Bridge A would always use the shortest path to reach every other bridge in the network. However, traffic entering Bridge D that is destined for Bridge C must traverse the path through Bridge A to reach Bridge C, even though Bridge D is directly connected to Bridge C. Clearly the path from Bridge D to Bridge C is not the shortest path in this case.

ISIS-SPB

The IEEE 802.1aq standard for SPB specifies the use of the IS-IS link state protocol instead of Spanning Tree to form sets of shortest path trees through the network. When SPB is used, each bridge is the Root for all traffic entering that bridge. As a result, each bridge can provide the shortest path to every other bridge in the network.

The bridging methodology needed to allow each bridge to serve as its own root bridge is enforced through the use of SPB BVLANs. This type of VLAN does not learn customer MAC addresses or flood unknown unicast and multicast traffic. In addition, network loops are mitigated through strict ingress checks based on the source MAC address of frames received on the BVLAN (frames received from an unexpected source are discarded).

SPBM uses an extended version of the IS-IS protocol that supports SPB (ISIS-SPB) to calculate the SPBM network topology. In addition, the learning and propagation of source MAC addresses is handled through the ISIS-SPB control plane, instead of through the data plane.

When calculating the SPBM network topology, ISIS-SPB must meet Layer 2 requirements to create congruent and symmetric paths. To do this, SPBM supports 16 predefined Equal Cost Tree (ECT) algorithms to break ties when two or more equal cost paths to the same destination are discovered. The same ECT algorithm is configured for the same BVLAN ID on each SPB switch in the network to ensure congruent, symmetric paths for the service traffic bound to that BVLAN.

Basically, to create a unicast tree, SPBM simply computes the shortest path from every bridge with each bridge serving as the Root (as shown below) and populates the Layer 2 forwarding database (FDB) on the SPB bridges with MAC addresses.

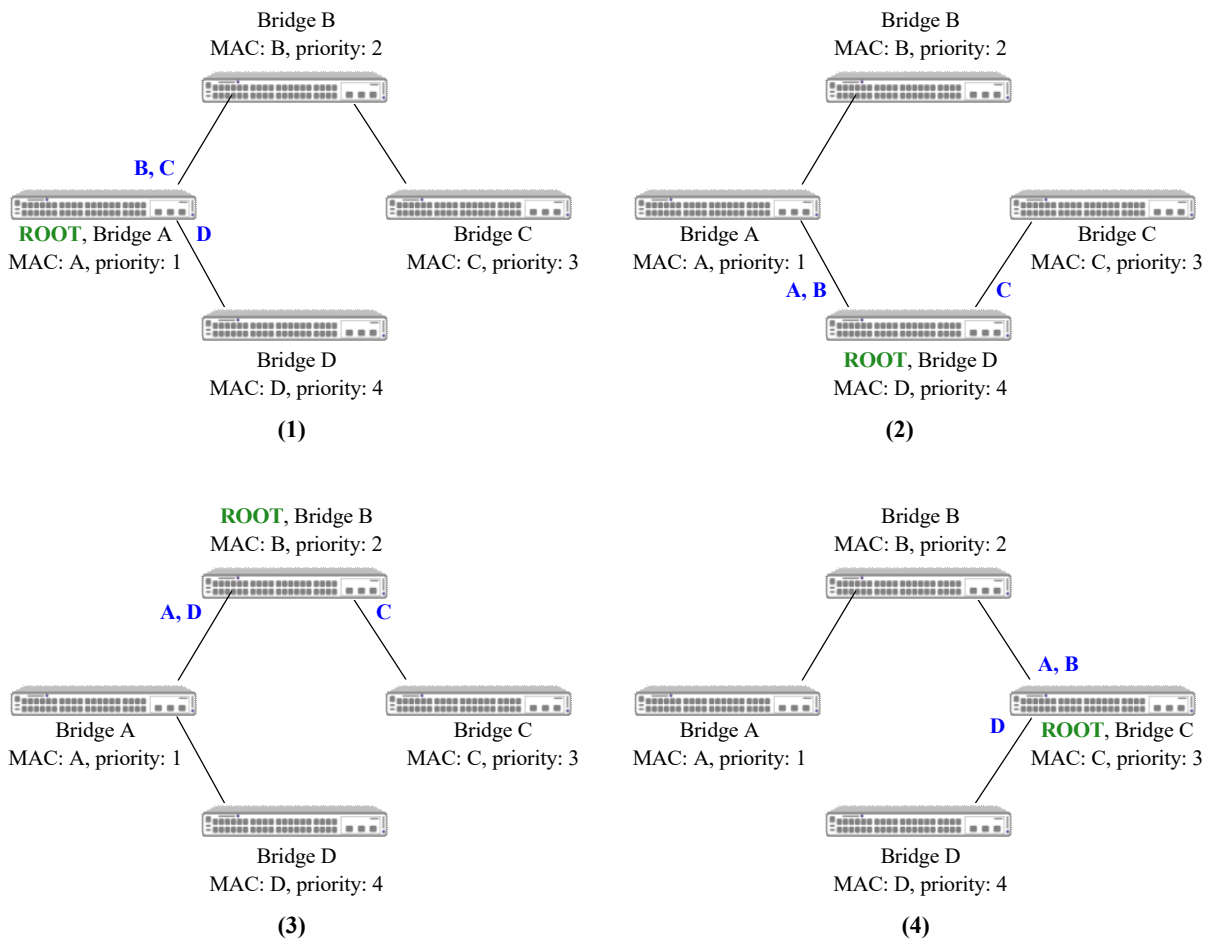


Figure 7-3 : ISIS-SPB Shortest Path Calculations

The ISIS-SPB unicast trees shown in Figure 3 were built as follows:

1 Bridge A calculates the shortest path tree to Bridge B and then programs its FDB with MAC address B on the link, as shown in (1).

2 Bridge A will then calculate shortest paths to Bridge C and Bridge D and programs the MAC addresses according to the path computed.

3 All other bridges follow the same procedure (note that the actual computation is much more optimized and the description here is only for illustration purposes).

4 The following traffic pattern for this example network is the result of the ISIS-SPB SPT calculations:

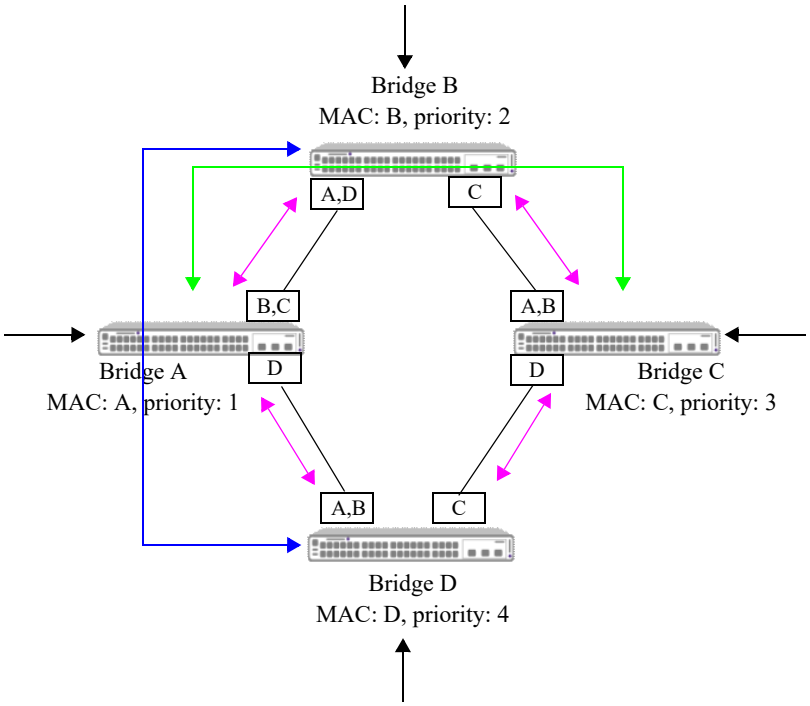


Figure 7-4 : ISIS-SPB Topology

As shown in Figure 4, all the backbone MAC (BMAC) addresses are learned by the switches when ISIS-SPB converges. The path taken by each unicast flow (for example ABC, CBA) are reverse path congruent and travel the shortest path through the network.

In the ISIS-SPB topology (Figure 4), the link between Bridge D and Bridge C carries traffic, whereas in the Spanning Tree topology (Figure 3), this link is blocked. Although these examples are based on traffic distribution for a single BVLAN, the ability to make all links in the topology available at all times is especially advantageous in highly redundant, meshed networks.

Although the link between Bridge D and Bridge C is used in the ISIS-SPB topology, traffic flow is relatively low in comparison to the other links. To make better use of this link, a second BVLAN could be created and assigned a different ECT algorithm to trigger ISIS-SPB calculations of a separate set of SPTs for the second BVLAN. This is similar to creating a new Multiple Spanning Tree (MST) instance in a Spanning Tree topology to create a different tree and assigning a new VLAN to that instance.

Each ECT algorithm uses a different calculation to break ties when paths between SPB bridges are equal cost. Another method to influence the SPT calculation is to modify the bridge priority for the switch or change the link cost metric for the SPB interface connection between two switches.

Multicast Traffic

SPBM supports two methods for replicating and forwarding multicast traffic (or unknown destination traffic) received from customer equipment: head-end replication and tandem replication.

- **Head-end replication**. Multicast traffic is replicated once for each receiver, encapsulated with the BMAC address, and then sent as a unicast packet to each destination. This method is more suited for networks where there is a low demand for multicast traffic.
- **Tandem replication**. Multicast traffic is replicated only where there is a fork in the SPT and each branch has at least one receiver. Each multicast source bridge in the SPBM network is the root for a

multicast distribution tree (MDT). An MDT is created per-source per-BVLAN and it is pruned according to whether the SPB node is on the shortest path of a multicast transmitter and receiver. For those MDTs that cross a given Backbone Core Bridge (BCB), that BCB needs to generate a multicast forwarding table for each such MDT.

Multicast traffic originating from a bridge is encapsulated with a special multicast BMAC DA that identifies the source of the traffic and then forwarded on the tree. Participating bridges that receive the packet will then know the source of the traffic and will use the multicast forwarding information for that source to switch the packet to the appropriate destination.

SPB Services

The SPBM network topology consists of two layers:

- **The backbone infrastructure (control plane) layer**. ISIS-SPB builds the backbone layer by defining loop-free, shortest path trees (SPTs) through the backbone network (see "SPBM Shortest Path Trees" on page 7-7 for more information).
- **The services (data plane) layer**. The service layer is based on the Provider Backbone Bridging (PBB) framework as defined in the IEEE 802.1ah standard. SPBM supports the 802.1ah MAC-in-MAC method for data encapsulation. SPBM services transport the encapsulated traffic over the ISIS-SPB infrastructure.

The SPB service layer framework is comprised of the following components:

- **Backbone Edge Bridge (BEB)**. An OmniSwitch is considered a BEB if the switch is SPB capable and at least one service access point (SAP) and one SPB interface is configured on the switch. The BEB marks the boundary between the customer network and the PBB network (PBBN).
- **Backbone Core Bridge (BCB)**. An OmniSwitch is considered a BCB if the switch is SPB capable and no SAPs are configured but at least one SPB interface is configured on the switch to forward encapsulated SPBM network traffic. Note that the requirement for configuring a BCB is based on whether or not the network topology includes a transit bridge.
- **Service Instance Identifier (I-SID)**. Configured only on a BEB, this component identifies a backbone service instance that will tunnel the encapsulated data traffic through the PBBN between BEBs. The I-SID is bound to a BVLAN ID and a Service Manager SPB service ID when the service is created.
- **Access Port**. A port or link aggregate configured as an SPB access port. This type of port is configured on the BEBs and defines the point at which traffic from other provider networks or directly from customer networks enters the PBBN. The access port is also associated with a Layer 2 profile that specifies how to process protocol control frames received on the port
- **Service Access Point (SAP)**—A SAP is a logical service entity (also referred to as a virtual port) that is configured on a BEB to bind an access port to an SPB service ID and specify the type of customer traffic ((untagged, single-tagged, double-tagged, or all) to encapsulate and tunnel through the PBBN.
- **SPB Interface (Network Port)—**A port or link aggregate configured as an SPB interface that resides on either a BEB or a BCB and connects to the backbone network. Network ports carry customer traffic encapsulated in 802.1ah frames and are associated with all BVLANs on the switch. Customer traffic ingressing on a network port is switched to another network port (on BCBs) or to an access port (on BEBs).

Once the ISIS-SPB infrastructure and the SPB service-based architecture is defined, the following service components are dynamically created by the OmniSwitch. No user-configuration is required.

- **Service Distribution Point (SDP)**—An SDP provides a logical point at which customer traffic is directed from one BEB to another BEB. SDPs are used to set up distributed services, which consist of at least one SAP on a local node, one SAP on a remote node, and an SDP binding the service on both nodes.
- **SDP Bind**—An SDP binding represents the binding of an SPB service instance to an SDP. The SDP then distributes the service connectivity to other BEBs through the ISIS-SPB shortest path trees.

Sample SPBM Network Topology

The following diagram provides a sample SPBM network topology that shows how the SPBM service and ISIS-SPB backbone layers work together to basically extend (or virtualize) customer traffic across a Provider Backbone Bridge Network (PBBN):

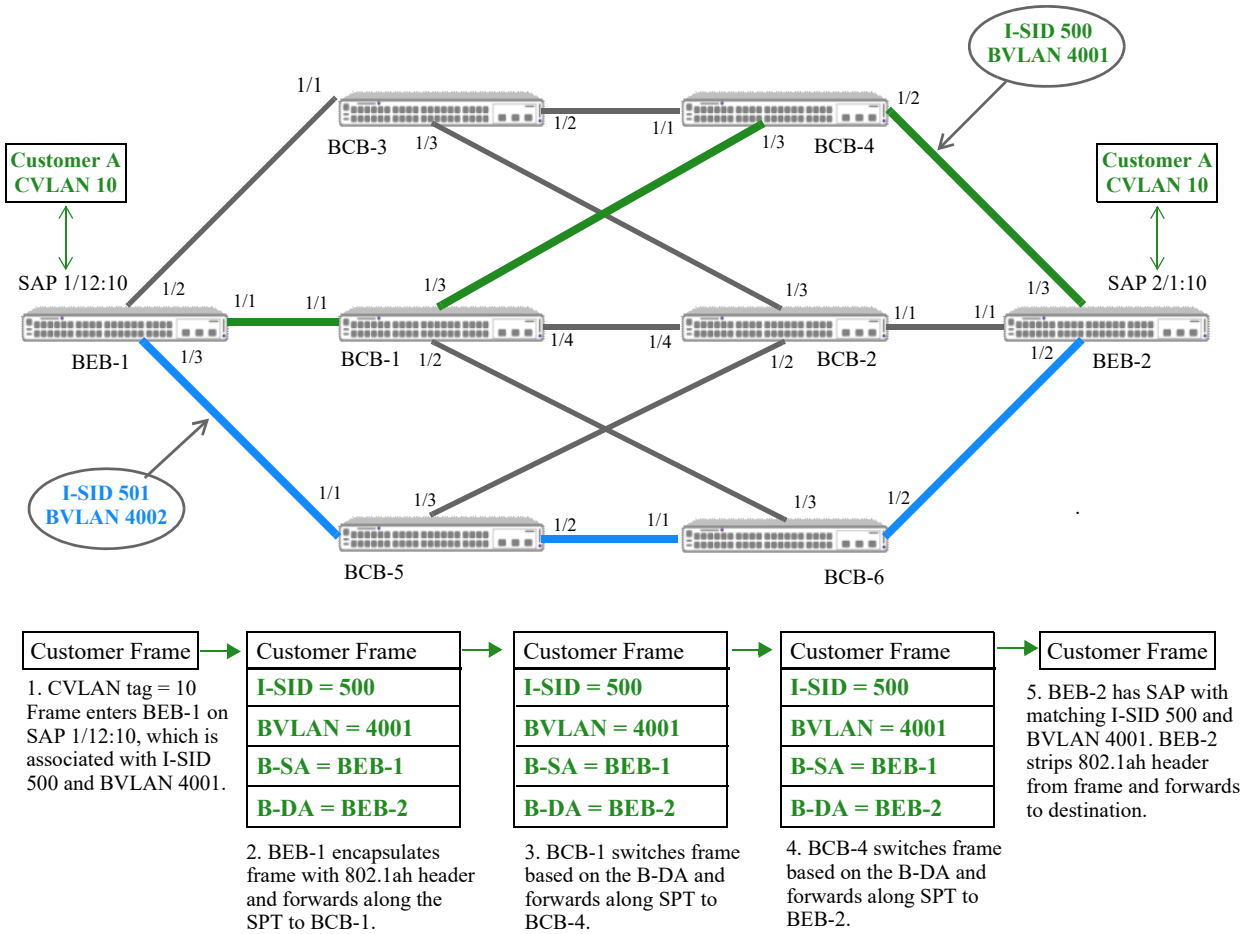


Figure 7-5 : Sample SPBM Network

In this sample SPBM topology:

- The packet flow for Customer A frames tagged with VLAN 10 is shown as a typical example. These frames are mapped to an SPB service that represents a binding of I-SID 500 to BVLAN 4001. The path for this binding is shown in green.
- An additional path, shown in blue, is for another SPB service that represents a binding of I-SID 501 to BVLAN 4002. This provides an example of how adding an additional BVLAN and service

configuration to the topology can provide an alternate service path for other traffic from the same customer or traffic from a completely different customer.

- SPB BVLAN 4001 and 4002 are created and assigned to ECT ID 1 and 2, respectively, on every switch (BEBs and BCBs) in the topology. These BVLANs serve as the transport entity on which ISIS-SPB builds the shortest path trees and SPB services tunnel data.
- The switch ports connecting each SPB switch with the next-hop SPB switch are configured as SPB interface ports. This type of port is used to forward ISIS-SPB control packets and serves as a network port for tunneling encapsulated traffic through SPB services.
- The service access points (SAPs) created on BEB-1 and BEB-2 determine which frames from Customer A are accepted on the SAP port, where they are then encapsulated and mapped to the associated service. Other SAPs exist on these switches for the other service path.
- When a frame tagged with VLAN 10 ingresses on port 1/12, the frame is encapsulated in an 802.1ah header. The header specifies the BMAC for BEB-1 as the B-SA, the BMAC for BEB-2 as the B-DA, the SAP I-SID (500), and the SAP BVLAN (4001).
- All other frames ingressing on SAP 1/12:10 that are not tagged with VLAN 10 are dropped, unless there are other SAPs configured for that port that will classify those frames.
- The encapsulated frame is then forwarded along the BVLAN 4001 shortest path tree (SPT) to BEB-2, where the 802.1ah header is stripped off and the frame is forwarded to the appropriate destination port.
- The entire process for encapsulating and tunneling customer frames is the same for frames ingressing on port 2/1 of BEB-2 destined for BEB-1.

How it Works

- There is one instance of ISIS-SPB supported in the backbone topology. This instance is activated once the BVLANs and SPB interfaces are created and the administrative status of ISIS-SPB is enabled for each switch.
- When ISIS-SPB is administratively enabled on each switch, all the configured SPB interfaces start to advertise Hello packets to discover and establish adjacencies with other SPB switches.
- Once adjacencies are established, link state packets (LSPs) are generated with SPB-specific TLVs and shortest path trees from each switch to all other switches are calculated.
- Each SPB switch learns the backbone MAC (BMAC) address and associated BVLAN IDs of every SPB switch in the network and stores that information in a local forwarding database. The BMAC address is the bridge MAC address of the switch and is advertised by ISIS-SPB as the System ID.
- ISIS-SPB then informs Service Manager of the reachability of the BMAC/BVLAN combinations. This information is used to automatically create a service distribution point (SDP) between the same BVLAN on each BEB.
- When ISIS-SPB receives advertisement of a service instance identifier (I-SID) from a remote BEB that matches an I-SID created on the local switch, the SDP (BMAC/BVLAN) of the remote BEB is bound to the I-SID. The binding of a service to an SDP is referred to as a mesh SDP.
- Basically, an SDP is a dynamically created logical entity that distributes service connectivity to other BEBs through the ISIS-SPB shortest path trees.When customer frames are then classified into a specific SAP, the frames are encapsulated and tunneled through the mesh SDP (service/SDP bind) associated with that SAP.

For CLI configuration examples, see “Quick Steps for Configuring SPBM” on page 7-36.

SPBM Pseudo-Wire (E-LINE Transparent) Service

The OmniSwitch implementation of Shortest Path Bridging provides customers at various sites with a multipoint-to-multipoint (E-LAN) connectivity solution. When an SPB backbone service instance (I-SID) is created on a BEB, SDP tunnels are automatically created with other BEBs on which the same I-SID is configured. This results in a full mesh of connectivity between the BEBs that provides any-to-any connectivity for the service traffic on the I-SID. However, as the number of users increases, so does the number of MAC addresses that are dynamically learned to minimize the amount of traffic flooding in the network.

To help reduce the use of system resources and prevent MAC address explosion, an SPB service can be configured as a pseudo-wire type of service to provide a single point-to-point (E-LINE) connection between two SAP attachment points. The attachment points to customer edge (CE) devices can be between two local SAPs associated with the same service or between two SAPs across the SPB network, as shown in the following diagram:

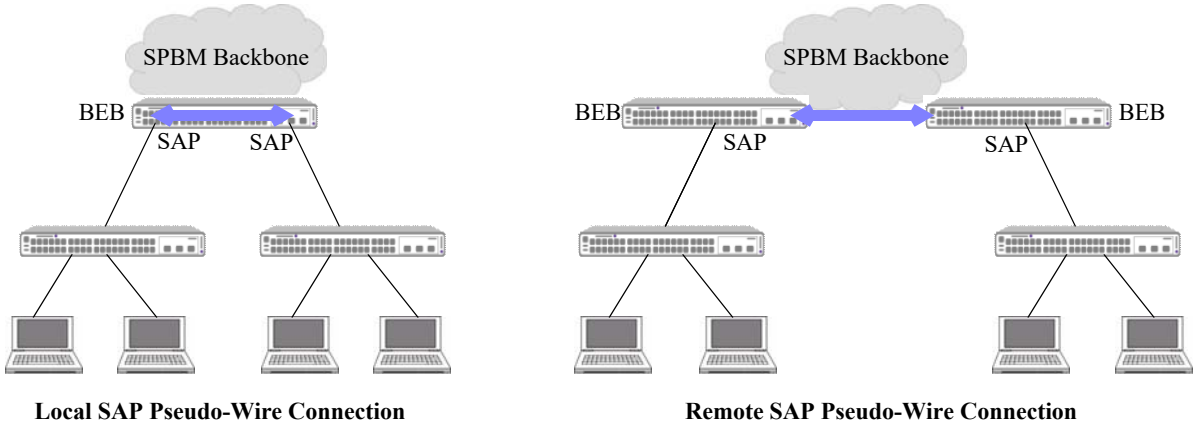


Figure 7-6 : Local and Remote SPB Pseudo-Wire Connections

An SPB pseudo-wire service reduces the number of customer MAC addresses learned and simplifies the flow of user traffic, as follows:

- Packets are transparently forwarded (MAC addresses are not learned) between two local SAPs.
- Packets from one SAP are encapsulated and transparently forwarded out of the SPB service network port.
- The encapsulation is removed from packets received from the SPB network port and are transparently forwarded out of the SAP.
- MAC address learning is automatically disabled, since there is no forwarding decision to be made. Packets entering one SAP attachment point will simply egress the other SAP attachment point of the pseudo-wire unchanged.
- Flooding and replication is not necessary since only two virtual ports are involved (SAP to SAP for a local service or SAP to a network port for the same service instance across the SPB backbone).

Note. When a pseudo-wire service connects two remote SPB nodes across the SPB backbone network, the ISIS-SPB control protocol still advertises the I-SID of the pseudo-wire service to each node to maintain the virtual connectivity between the two nodes.

For more information and configuration examples, see "Configuring an SPB Pseudo-Wire Service" on page 7-63 and "SPB Pseudo-Wire Configuration Example" on page 7-64.

Remote Fault Propagation for SPBM Services

When a point-to-point connection is emulated with a Layer 2 SPB service, it is necessary to propagate connectivity faults from one end of the service tunnel to the other end. This allows a locally connected device to detect a connectivity fault in the SPB service and take action (such as enable a redundant link or send a trap) in response to the detected fault. Remote Fault Propagation (RFP) for SPB provides this type of fault detection and propagation from one end of an SPB service to the other.

The RFP functionality is applied to the SPBM service (data plane) layer. Connectivity fault events are propagated into an SPB Service Access Point (SAP). A SAP is associated with an SPB access port and a service instance identifier (I-SID). When a SAP port goes down, the SAP port on the other end of the service is also brought down. Without the RFP for SPB feature, the other end would continue to transmit packets waiting for a response.

Ethernet OAM messaging is used to detect a failed condition and propagate the fault. An OAM Continuity Check Message (CCM) is sent at specified intervals between SAPs to advertise the status of SAP components (such as the SPB access port and I-SID information).

This implementation of RFP for SPB involves setting up the following components:

- An underlying SPB network infrastructure. RFP will monitor SPB access ports, which are bound to SAPs. A SAP consists of an access port, SPB service ID, and an encapsulation value (the VLAN tags that the SAP will process on the access ports).
- An RFP domain, which consists of local maintenance end points (MEPs) with remote end point lists that are assigned to the same RFP domain ID.
  - A local MEP defines the RFP domain parameters, such as the RFP domain ID, level, and CCM interval. An ID number is assigned to the local MEP to identify the local switch as a participant in an RFP OAM domain.
  - A remote end point list identifies the SPB services to monitor and the remote end points (the MEP IDs of remote switches) to which the status of the services is advertised. Configuring the remote end point list of an RFP domain triggers the sending of CCM packets.
- A reserved Ethernet OAM domain to which the RFP domain is mapped. When the local MEP of an RFP domain is configured, an OAM domain is automatically created based on the parameters specified when the local RFP MEP was created.

The following diagram shows how RFP works in a sample SPBM network topology:

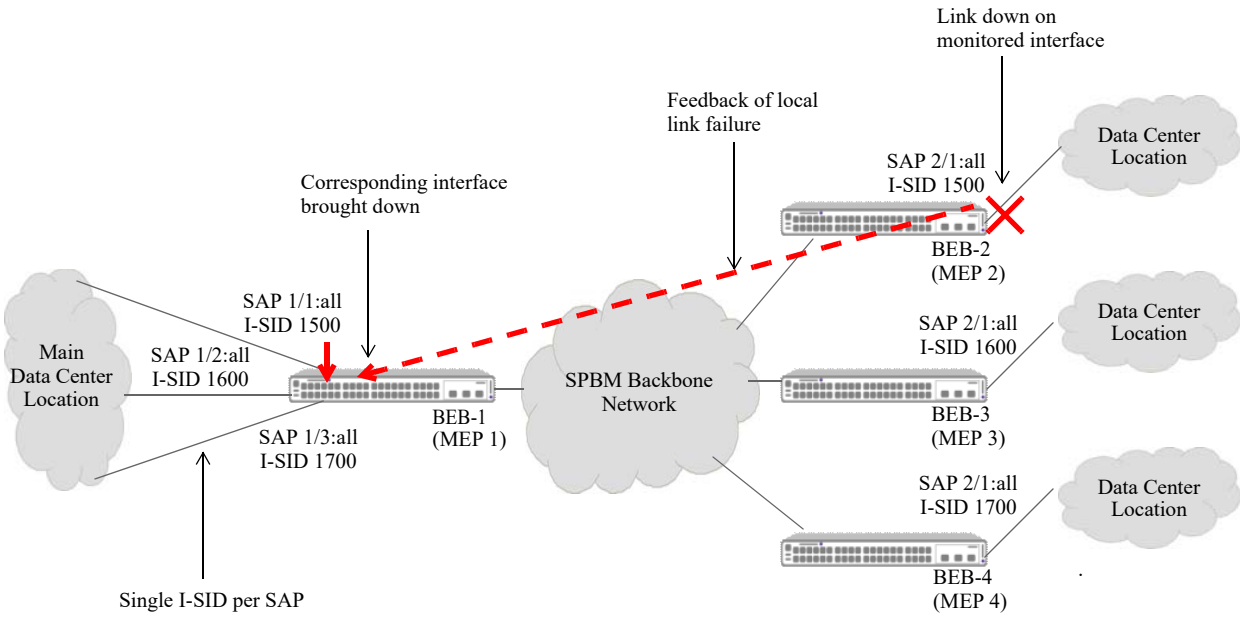


Figure 7-7 : RFP in a Sample SPBM Network

In this sample SPBM topology:

- An RFP local MEP and a remote end point list are configured on each Backbone Edge Bridge (BEB). Both are assigned to the same RFP domain ID on each BEB to identify the end points as participating members of the RFP domain.
- Each local MEP is assigned an ID number, which is used as the virtual UP MEP ID. In this example, the virtual UP MEP ID is 1 for BEB-1, 2 for BEB-2, 3 for BEB-3, and 4 for BEB-4.
- Each remote end point list specifies the SPB services to monitor and the MEP IDs of remote BEBs to which the status of the services is advertised. For example, the remote end point list on BEB-1 contains the monitored SPB services and local MEP IDs for BEB-2, BEB-3, and BEB-4.
- The remote end point list binds an SPB service ID to the RFP domain. The service ID is associated with a service instance identifier (I-SID) and a SAP, which identifies the SPB service instance and access port to monitor. For example, on the BEB-2 switch, the status of I-SID 1500 on access port 2/1 is monitored and advertised to BEB-1.
- CCM packets transmitted on the RFP domain advertise I-SID and access port status information for the local SAP. The SAP information to advertise is identified through the SPB service ID that is associated with the RFP domain. For example, BEB-1 and BEB-2 both advertise the status of SAP port 1/1 and SAP port 2/1 for I-SID 1500. The same SPB service ID is mapped to each of these SAPs, which means the same I-SID is mapped to each of these SAPs.
- When port 2/1 goes down on BEB-2, the service represented by I-SID 1500 stops transmitting. The CCM packets transmitted between BEB-2 and BEB-1 detect and advertise the port down fault. This causes BEB-1 to administratively down port 1/1 in response to the fault condition.

For an example of the CLI configuration for this sample deployment of RFP in an SPB network, refer to the"RFP for SPB Configuration Example" on page 7-70.

Customized CCM Packets

The CCM packets transmitted between RFP end points contain a proprietary OUI TLV that provides link fault information for the SPB services that are monitored by the RFP domain. The following shows an example of the proprietary TLV format:

Type = 127	Length	OUI MAC == Alcatel OUI (3 octet) Information about ISID, Portstate ISID 24 bit value (3 octet) Port state (UP/STATE) (1 octet) The above information is repeated if there are multiple I-SIDs.

- OAM unicast CCM packets are sent without Provider Backbone Bridge (PBB) header encapsulation across the SPBM network to each remote BEB device on the control B-VLAN.
- Only information related to the I-SID associated with the remote BEB is sent in the proprietary TLV.
- OAM packets are filtered on the SPB SDP interfaces to capture only the CCM packets used for RFP monitoring.
- Only CCM information will be processed for the related I-SID information on the receiving switch.

Fault Detection

Each BEB in the RFP domain will check the I-SID and port state information contained in the received CCM packets.

- If any port state has transitioned from up to down, the local SAP port associated with the same I-SID is also brought down as a port violation.
- When a CCM indicates that the downed port has transitioned back to an up state, the local port violation is cleared.
- After a port violation is cleared, a 10 second timer is started to avoid bringing down the local ports immediately. This allows for the scenario in which a port violation is manually cleared on one BEB and by the time the violation is cleared on another BEB, a CCM packet from the other BEB is received with SAP port down information.
- If a BEB device goes down, the information about the BEB will time out on remote BEB devices after 3 multiplied by the value of the CCM interval (3*CCM interval value). For example, if the CCM interval value is set to 100ms, a remote BEB will wait 300ms before timing out the information about the BEB that went down. The local physical SAP access port mapped to the I-SID that timed out is then brought down as well.

For more information and CLI configuration examples, see "Configuring Remote Fault Propagation for SPBM" on page 7-66 and "RFP for SPB Configuration Example" on page 7-70.

For more information about Ethernet OAM, see Chapter 39, "Configuring Ethernet OAM."

IP over SPBM

The OmniSwitch implementation of SPBM provides L2 VPN capability that bridges L2 customer LAN segments. Customer edge (CE) devices form peers and exchange routing information, as well as perform the necessary IP forwarding. Then the SPBM BEBs bridge the already routed IP traffic across the SPBM backbone.

In addition to L2 VPN, the OmniSwitch also provides an IP over SPBM capability that consolidates the routing functionality of CE devices into the BEB devices. The Virtual Routing and Forwarding (VRF) instances on different BEBs are tied together via backbone I-SIDs across the same SPBM backbone that is used to support Layer 2 VPNs.

The OmniSwitch IP over SPBM solution supports two methods for combining L3 routing and L2 SPBM in the same switch: VPN-Lite and L3 VPN.

Note. The term “IP over SPBM” refers to both IPv4 and IPv6 over SPBM. If there are any differences between the implementation of IPv4 over SPBM and IPv6 over SPBM, an explicit reference to IPv4 or IPv6 is made.

VPN-Lite

The VPN-Lite method provides a gateway between a regular SPBM service and a router within the same OmniSwitch chassis. This solution provides a specific advantage in that it allows a single box to represent two tiers in a typical fat-tree network, which is popular in data center deployments.

In addition, a VPN-Lite configuration can act purely as an L3 VPN when configured correctly. In this mode, existing routing protocols can form adjacencies across the SPBM PBB network. To keep it purely an L3 VPN, the administrator makes sure that no SPBM SAPs that can inject bridged flows are allowed to attach to the I-SID designated for the specific VPN.

The VPN-Lite approach uses the SPBM network in the same way a VLAN is used for transporting L3 frames. Each BEB or host can inject frames into the I-SID as needed, and BEBs can decide to bridge or route those frames based on their inner and outer destination MAC address.

L3 VPN (ISIS-SPB)

When the L3 VPN method is implemented, the OmniSwitch acts as an access or edge router to multiple VRFs and connects these VRFs across an SPBM PBB network. Each VPN is identified by a local VRF instance on each BEB and globally in the backbone by an I-SID in the PBB header. ISIS-SPB will import and export routes from the local routing protocols running inside their respective VRFs. In essence, ISIS-SPB is creating tunnels between BEBs through which routed frames are sent to reach their target networks.

The OmniSwitch L3 VPN solution is based on the IETF drafts *IP/IPVPN services with IEEE 802.1aq SPB(B) networks* and uses IS-IS TLVs to exchange routes between the BEBs that host the same VPN services. This approach also gives an administrator the ability to build VPNs and extend them over an SPBM core.

L3 VPN Interface

An L3 VPN interface serves as an IP gateway to access remote networks and is required when using either the VPN-Lite or L3 VPN method.

Supported L3 VPN Interface Options

The following available options for defining an L3 VPN interface are based on the switch platform.

Switch	Inline Routing Service-Based (Single Pass)	Inline Routing Front-Panel Ports (Two Pass; No Cable)	External Loopback Cable (Two-Pass)
OmniSwitch 6860	No	No	Yes
OmniSwitch 6860N*	Yes	No	Yes
OmniSwitch 6865	No	No	Yes
OmniSwitch 6900	No	No	Yes
OmniSwitch 6900-V72/C32	No	Yes	Yes
OmniSwitch 6900-X48C6/ X48C4E/T48C6/V48C8/C32E*	Yes	No	Yes
OmniSwitch 9900	Yes	No	Yes

* The availability of switch resources (such as the number of Layer 3 interfaces, next hops, etc.) is reduced to support service-based inline routing. See "L3 VPN Interface: Inline Routing (Service-Based IP Interface)" on page 7-75 for more information.

L3 VPN Interface: Inline Routing

The following methods for defining an L3 VPN inline routing interface are supported:

- **Service based inline routing.** This method processes encapsulated SPB and IP packets in a single-pass through an IP interface that is bound to an SPB service. The L3 VPN interface is defined by configuring an IP service-based interface within a VRF instance. This connects routes within the VRF instance to an SPB service instance (I-SID).
- **Front-panel port inline routing**. This method utilizes front-panel ports to implement two-pass processing of encapsulated SPB and IP packets. The L3 VPN interface is defined by configuring a loopback port that will forward traffic between an SPB service and VLAN domain. Loopback ports can also be combined into a static link aggregate.

Using the inline routing mechanism simplifies the configuration in that a physical cable is not required to form an external loopback.

The following diagram shows a logical depiction of an inline routing configuration that defines an L3 VPN service-based interface:

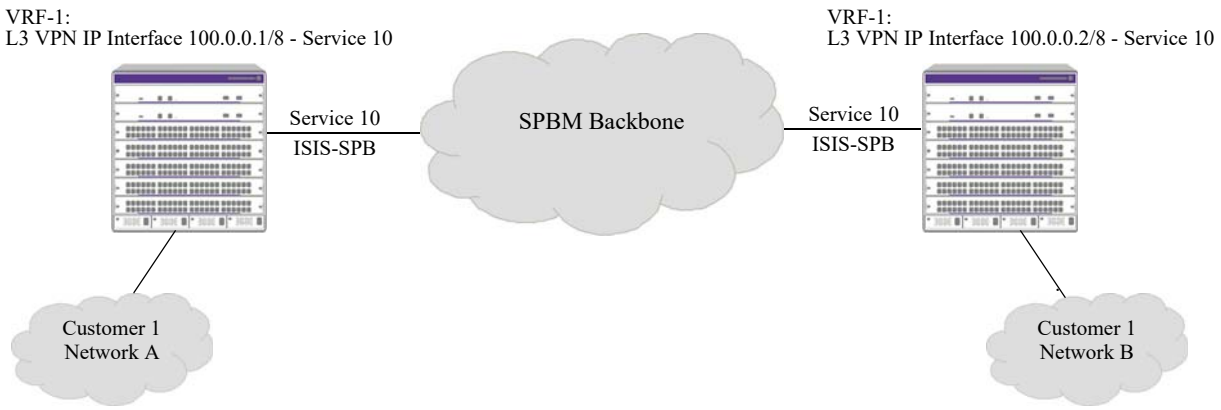


Figure 7-8 : L3 VPN Interface: Inline Routing (Service-Based IP Interface)

The following diagram shows a logical depiction of an inline routing configuration that defines an L3 VPN front-panel port interface:

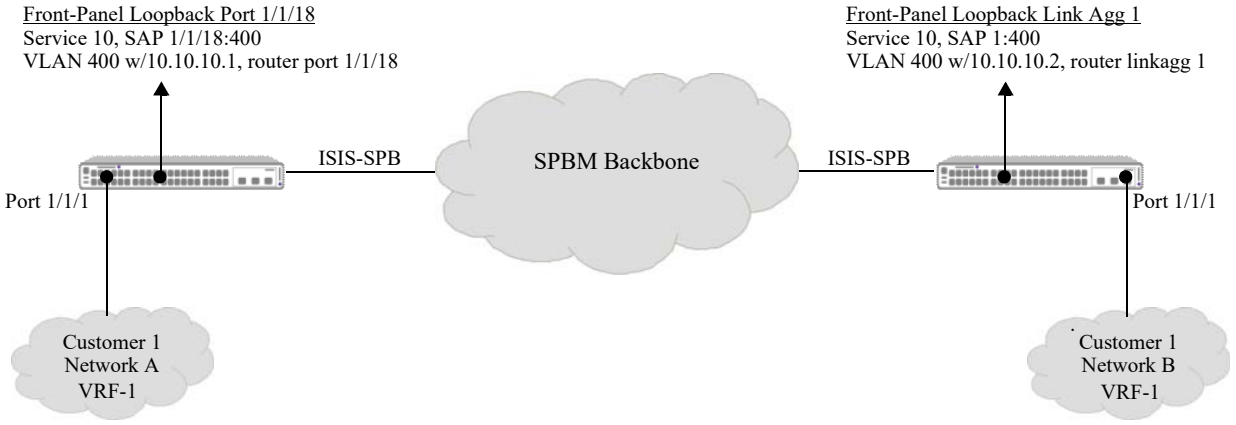


Figure 7-9 : L3 VPN Interface: Inline Routing (Front-Panel Ports)

L3 VPN Interface: External Loopback

External loopback is a two-pass processing mechanism that requires a physical cable to connect a regular port to a service access port. The regular port is tagged with an IP interface VLAN; the service access port is associated with an SPB Service Access Point (SAP). The VLAN-based IP interface serves as the L3 VPN interface.

A regular switch port or a static link aggregate can serve as a loopback port in an external loopback configuration. In addition, multiple loopback port pairs are allowed and can be shared between different VRFs.

An L3 VPN loopback interface configuration consists of the following components:

- An IP interface created in a specific VRF instance and bound to a VLAN ID.
- A regular switch port or link aggregate tagged with the IP interface VLAN.
- A service access port that is assigned to an SPB SAP. The SAP encapsulation is configured with the VLAN ID that is tagged on the regular switch port or link aggregate.

- A physical cable that connects the VLAN port with the service access port to form the loopback configuration. The service access port handles Layer 2 bridging into the service domain and the VLAN port handles the Layer 3 routing into the VLAN domain.

The loopback configuration connects a VRF to an SPB SAP and can carry traffic from different VRFs tagged with different VLANs. The following diagram shows a logical depiction of an L3 VPN loopback interface:

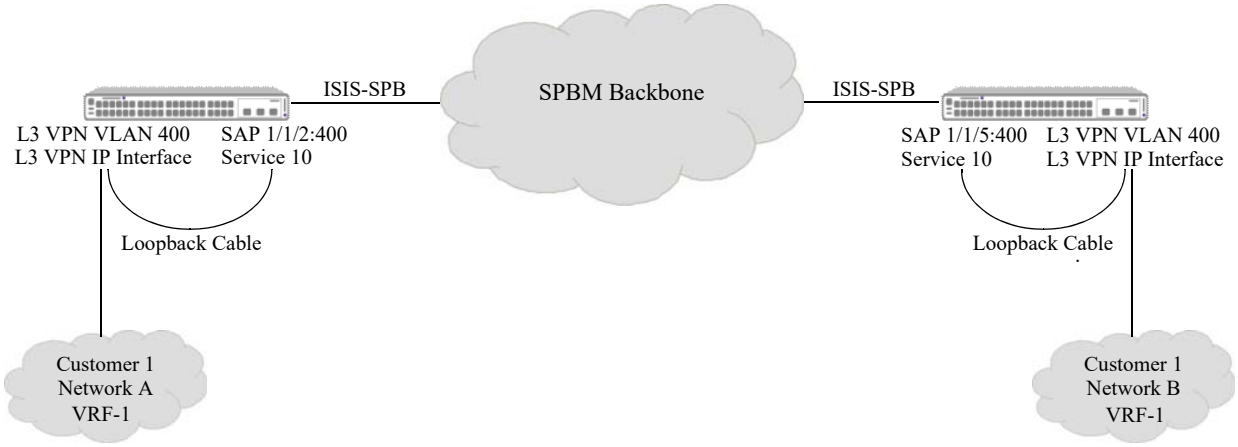


Figure 7-10 : L3 VPN Interface: External Loopback

How it Works

This section describes the VPN-Lite and L3 VPN control and data plane operations in an IP over SPB network configuration. Although both approaches use an L3 VPN interface configuration, they differ in how routing protocol control packets are exchanged and processed to support IP over SPB.

VPN-Lite Control Plane Operations

When routing protocols or static routes are running on the L3 VPN interface, the interface can exchange IP routes with other L3 VPN interfaces that are running the same routing protocols and are associated with the same I-SID. By exchanging routes with other L3 VPN interfaces, VRFs on different BEBs can learn remote networks from each other.

- If inline routing is used in this scenario, the routing protocol control packets are sent from the L3 VPN interface and carried on SDPs into the SPBM backbone. The control packets received from the SPBM backbone travel from SDPs to the VRF following the same process but in reverse.
- If a physical loopback port configuration is used in this scenario, the routing protocol control packets sent from the L3 VPN interface travel though the tagged VLAN port, enter the service access port, where the packets are then distributed into different services by SAPs associated with the access port and carried on SDPs into the SPBM backbone. The control packets received from the SPBM backbone travel from SDPs to the VRF following the same process but in reverse.

L3 VPN Control Plane Operations

The ISIS-SPB support of the IPVPN TLV, IPv4 sub-TLV, and IPv6 sub-TLV provides a different method for exchanging L3 routes between VRFs. Instead of running routing protocols on the L3 VPN interfaces, IP routes are imported into ISIS-SPB from VRFs. ISIS-SPB then carries these routes in the TLVs through the SPBM cloud to other SPBM BEBs. When ISIS-SPB receives IPVPN TLVs from the cloud, ISIS-SPB will export the routes to the appropriate VRFs.

The L3 VPN approach implements the importing and exporting of routes between ISIS-SPB and VRF instances and the transport of these routes using the supported TLVs. The administrator does not have to configure routing protocols on the L3 VPN interface. Implementing the L3 VPN approach requires careful consideration to avoid routing loops.

VPN-Lite and L3 VPN Data Plane Operations

Data is moved in the same manner for both VPN-Lite and L3 VPN traffic, and the existing data plane forwarding mechanisms for SPB and IP are used without modification:

- An L3 VPN interface serves as an IP gateway to access remote networks. The network administrator has to ensure the IP subnet reachability of the L3 VPN addresses on the same SPBM I-SID.
- IPv4 L3 VPN interfaces use dynamic ARP and IPv6 L3 VPN interfaces use neighbor discovery to learn the MAC addresses of other L3 VPN interfaces and provide next-hop forwarding information to the switch.
- IP data plane packets travel the same path as VPN-Lite control packets (see "VPN-Lite Control Plane Operations" on page 7-21 for more information).
- Data in the SPBM cloud is encapsulated into the Provider Backbone Bridge (PBB) format (see "SPB Services" on page 7-11 for more information).

For more information and configuration examples, see "Configuring IP over SPB" on page 7-73 and "IP over SPB Configuration Examples" on page 7-81.

SPB Over Shared Ethernet

By default, ISIS-SPB operates over point-to-point (P2P) links which allows only one adjacency on an SPB network interface. However, an SPB network interface can be configured to allow multiple adjacencies to form on the interface. This is particularly useful for extending an SPB backbone over a shared Ethernet domain, such as a service provider network or even connect to another ISIS-SPB domain.

An SPB multiple access (multi-access) network interface is configured on SPB Backbone Edge Bridges (BEBs) that connect directly to a shared network instead of to SPB Backbone Core Bridges (BCBs). Each BEB forms ISIS-SPB adjacencies over the shared network with all the other BEBs on the multi-access network interfaces.

Participating BEBs elect one of the multi-access network interfaces to serve as the Designated Intermediate System (DIS). The DIS represents all of the multi-access links as a virtual SPB node (pseudo-node).

Note. Software releases prior to AOS Release 8.7R1 do not process pseudo-node LSPs. As a result, SPB nodes running such software may experience inconsistent connectivity to destinations beyond the shared Ethernet network segment. If such network reachability is desired, those SPB nodes must be upgraded to AOS Release 8.7R1.

The following diagram shows an example of an SPB backbone extended over a service provider network.

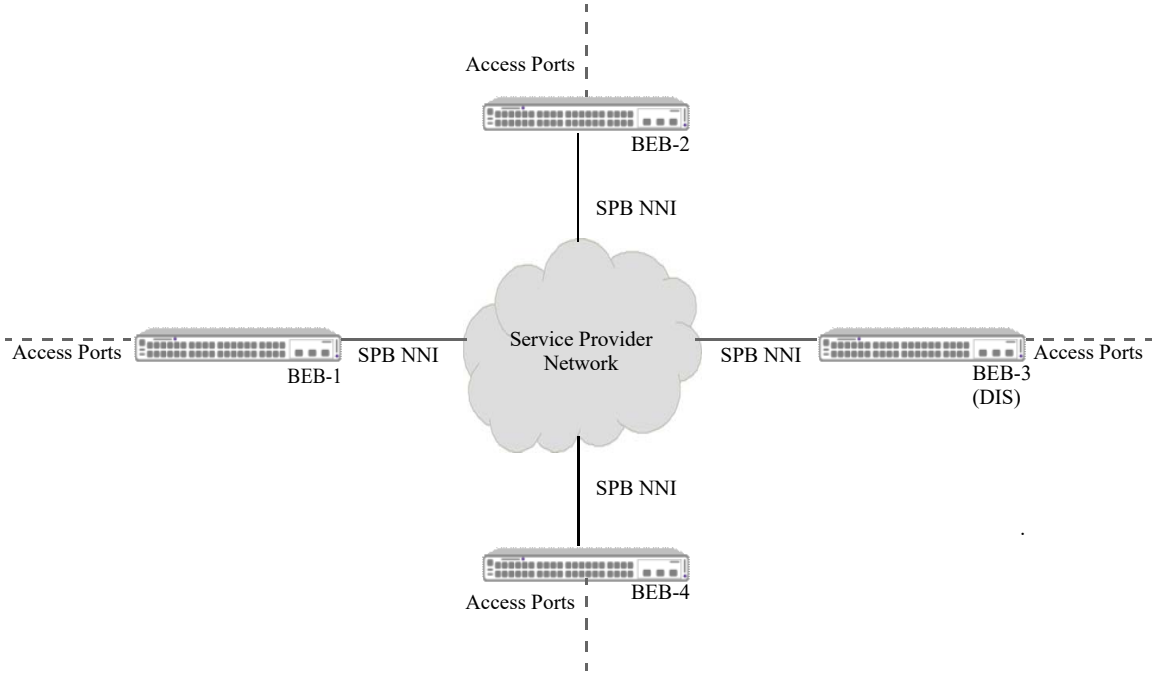


Figure 7-11 : SPB Backbone over a Service Provider Network

In order for the BEBs in this diagram to communicate, they need to form adjacencies with each of the other BEBs over the service provider network. This is not possible with a P2P configuration, so each SPB network interface port is configured as a multi-access LAN interface to allow multiple adjacencies to form across the broadcast network domain.

How it Works

The manner in which the SPB network backbone is built on multi-access interfaces is similar to how the network backbone is built on P2P interfaces. However, there are some differences.

- Multi-access interfaces exchange LAN Intermediate System-to-Intermediate System Hello (IIH) packets, which contain a priority value and a LAN ID (combination of System ID and a unique local Circuit ID); information that is not needed in P2P IIH packets.
- During the LAN IS-IS Hello packet exchange, an election process is triggered to select a Designated Intermediate System (DIS) for the LAN. Election of the DIS is based on the highest interface priority, which can also be manually configured. In case of a tie, the router with the highest backbone MAC address (BMAC) for that interface is elected as the DIS. This process is similar to how ISIS-IP operates in a multi-access, broadcast network.
- After adjacencies are formed and the DIS is elected, link state packets (LSPs) are exchanged. To ensure efficient LSP exchange and minimize bandwidth usage, all multi-access interfaces report their adjacencies to the DIS.
- The DIS represents all of the multi-access links as a virtual SPB node (pseudo-node) to IS-IS by generating a pseudo-node LSP. This type of LSP contains a list of all the switches connected to the multi-access network, including the DIS, that comprise the virtual SPB node.
- The LSP database synchronization over multi-access links differs somewhat from how the database is synchronized over P2P links as follows:
  - P2P links synchronize the LSP database by sending a Complete Sequence Number PDUs (CSNP) once when an adjacency is initialized and before LSP exchange begins. LSPs are then exchanged over P2P links in a way that ensures that all LSPs sent over the link from one end to the other are received.
  - With multi-access links, the DIS is responsible for synchronizing the LSP database across the shared network. On multi-access links, CSNPs are flooded periodically by the DIS to coordinate database synchronization.
- The DIS pseudo-node LSP is used to calculate the shortest path tree over the multi-access links. The regular LSP generated by each SPB node lists the pseudo-node as its neighbor on multi-access links; the pseudo-node LSP lists all the other nodes as it neighbors on a multi-access link. As a result, the shortest path between any two nodes on a multi-access link goes through the DIS.
- ISIS-SPB records a list of hops (SPB nodes) during the shortest path first (SPF) calculations from one node to the other. However, the SPB pseudo-node is not counted in the list of hops, so it is not used as a tie breaker for ECMP calculations.

To the rest of the network, the SPB multi-access links are seen as a virtual node that is defined and represented by the DIS through pseudo-node LSPs. The following diagram provides a logical depiction of how the SPB pseudo-node is interpreted by IS-IS with a DIS:

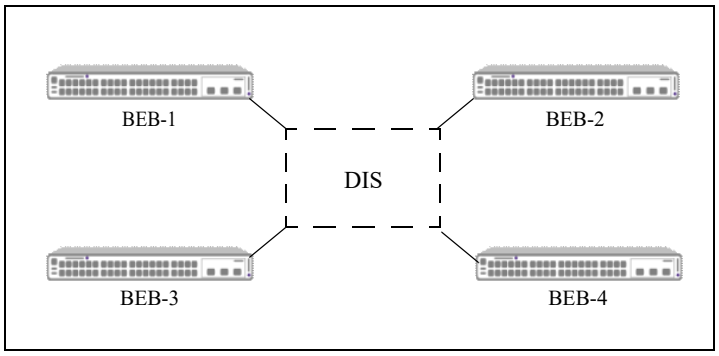


Figure 7-12 : Network Interpretation of SPB Pseudo-Node

For more information and configuration examples, see "Configuring SPB Over Shared Ethernet" on page 7-101 and "SPB Over Shared Ethernet Configuration Examples" on page 7-104.

SPB In-Band Management

In an SPBM network, Backbone Edge Bridges (BEBs) and Backbone Core Bridges (BCBs) can be directly managed within the SPBM bridging domain by configuring a single IP interface on a Control BVLAN. This avoids having to set up a separate VLAN domain to provide IP connectivity across the SPBM domain.

In addition to configuring an IP interface on a Control BVLAN, use one of the following methods to advertise the reachability of the management network to the SPB nodes:

- Configure static routes to route packets to destinations outside of the IP BVLAN subnet.
- Configure the redistribution of management network routes into ISIS-SPB at one or more BEBs that connect to a management station. ISIS-SPB then advertises the management network routes, along with the locally configured IP Looback0 address and IP BVLAN address, to all SPB nodes in the network.

The redistribution of management network routes into ISIS-SPB provides a simpler method for SPB in-band management by not requiring the configuration of static routes on every SPB node.

A BEB that connects to a management station on the customer side is referred to as a manager BEB. A managed SPB node is a BEB without management stations on the customer side or a BCB that does not access the customer side. The manager BEB serves as a gateway to the management networks.

When a managed SPB node receives the advertised management network route information, the network route and the IP Looback0 route are injected with the IP BVLAN address as the gateway into the management VRF instance as SPB management (SPB-MGMT) routes. The managed SPB node then knows how to reach the manager BEB. In turn, the managed SPB node advertises its Loopback0 address along with the already received IP BVLAN address, so that the manager BEB knows how to reach the managed SPB node.

The following diagram shows an example of advertising reachability to management networks, SPB nodes, and remote switches by redistributing routes into ISIS-SPB.

- The management network routes are configured or learned on BEB-1 and BEB-3. The routes are then redistributed into ISIS-SPB, which then propagates the management network route, the locally

configured IP Looback0 address, and the IP BVLAN address in an LSP that is sent to all SPB nodes in the network.

- The receiving SPB nodes send the advertised information to the management VRF routing table. Persistent ARP entries are then added to map the sender's system MAC address to the sender's IP BVLAN interface address. All SPB nodes can now be accessed by the management stations.
- In addition, the route to a remote switch network is redistributed into ISIS-SPB on BEB-3 and then propagated to all SPB nodes. The learned SPB-MGMT routes on BEB-3 can then be redistributed into the remote network so that remote switches can also be managed.

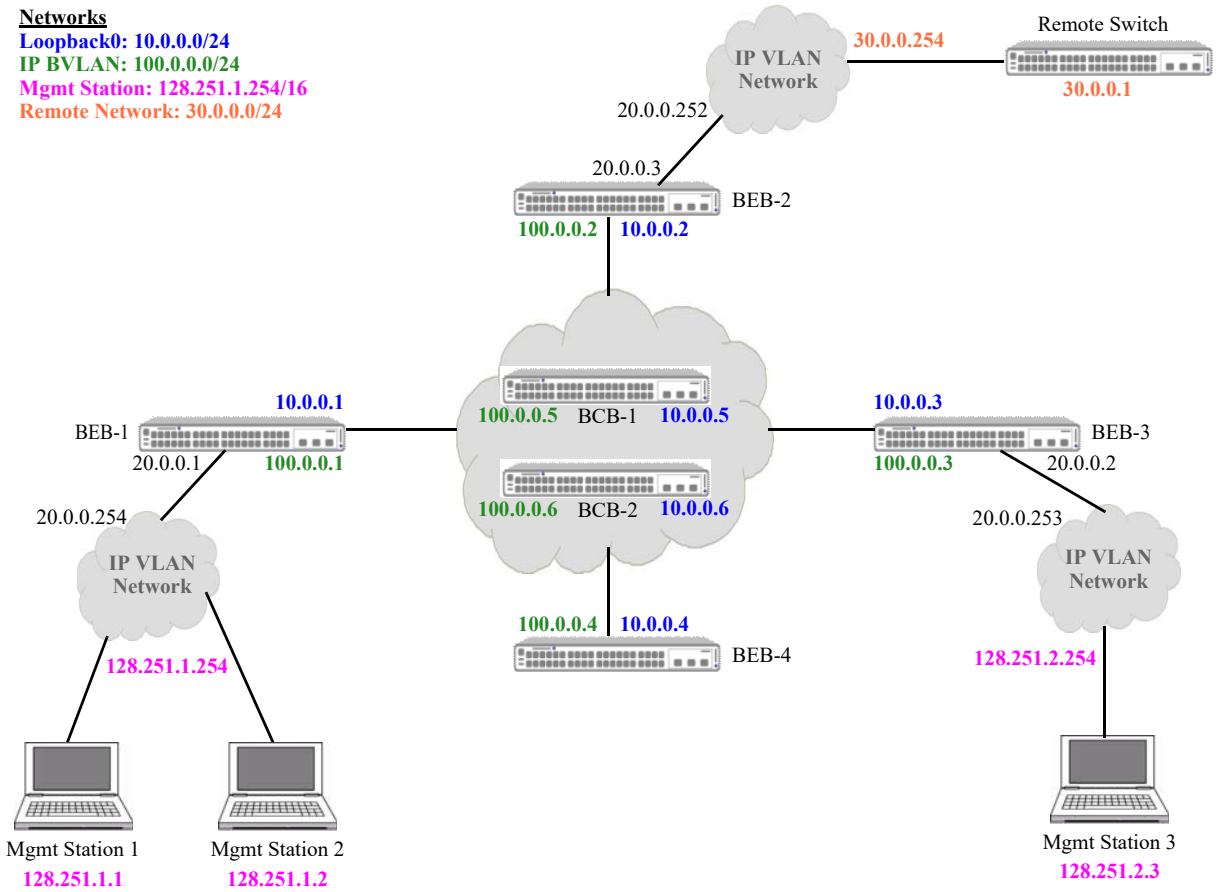


Figure 7-13 : SPB In-Band Management - Route Redistribution

In this sample SPB network configuration,

- On BEB-1, route 128.251.1.0/24 -> 20.0.0.254 is installed and redistributed into SPB. BEB-1 then advertises the following information to all of the other SPB nodes:
  - Management network route 128.251.1.0/24.
  - Loopback0 route 10.0.0.1/32.
  - BVLAN IP address 100.0.0.1
- On BEB-3, route 128.251.2.0/24 -> 20.0.0.253 is installed and redistributed into SPB. BEB-3 then advertises the following information to all of the other SPB nodes:
  - Management network route 128.251.2.0/24.
  - Loopback0 route 10.0.0.3/32.
  - BVLAN IP address 100.0.0.3

- When the two BCBs and all the other BEBs receive the information advertised by BEB-1 and BEB-3, the following occurs:
  - SPB routes 128.251.1.0/24 -> 100.0.0.1 and 128.251.2.0/24 -> 100.0.0.3 are installed.
  - Routes 10.0.0.1/32 -> 100.0.0.1 and 10.0.0.3/32 -> 100.0.0.3 are installed.
  - Static ARP entries are added to map 100.0.0.1 to the system MAC address of BEB-1 and 100.0.0.3 to the system MAC address of BEB-3.
- To manage the switch on a remote network accessed through BEB-2,
  - Route 30.0.0.0/24 -> 20.0.0.252 is installed and redistributed into SPB.
  - BEB-2 advertises the remote network route 30.0.0.0/24, the Loopback0 route 10.0.0.2/32, and the BVLAN IP address 100.0.0.2 to all of the other SPB nodes.
  - BEB-2 redistributes the learned SPB-MGMT routes into the remote network.

Configuring SPB In-Band Management

Consider the following guidelines when configuring SPB in-band management access:

- A single IPv4 interface can be configured on a Control BVLAN to provide in-band management access in the SPBM domain. When creating an IPv4 interface for a Control BVLAN, consider the following:
  - ISIS-SPB is the only protocol supported in the IP BVLAN domain for exchanging or advertising IP routing information. No other routing protocol (including VRRP) is supported. However, SPB-MGMT routes can be redistributed into dynamic routing protocols.
  - The IP interface is considered operationally active when the underlying Control BVLAN becomes operationally up and all the underlying configuration is considered valid.
  - Multi-netting of IP interfaces on a BVLAN is not supported. Only one IP interface per Control BVLAN is allowed.
  - STP is not allowed on BVLANs, and all broadcast packets will be restricted on this IP interface. As a result, explicit ARP/ND resolution is not supported. ISIS-SPB will provide the MAC-to-IP address mappings to avoid broadcast packets in the SPBM backbone domain.
  - The IP BVLAN domain will operate in IP unicast mode; IP Multicast is not supported.
- Any VRF instance can be used to redistribute management routes into ISIS-SPB. However, only one VRF can be used and it must be the VRF where the management IP interface resides.
- SPB management route information is represented as the protocol SPB-MGMT, which is used as a source and destination protocol for route redistribution. For example, **ip redist spb-mgmt into ospf** or **ip redist ospf into spb-mgmt**. In addition, **show** commands used to verify IP routing information and display SPB LSPs provide an **spb-mgmt** parameter or include SPB-MGMT information (see "Verifying the In-Band Management Configuration" on page 7-29).
- SPB-MGMT routes are assigned a default route preference of 150, which is configurable using the **ip route-pref** command. Make sure this value does not conflict with the route preference configured for any other protocols.

Note. When using ISSU to upgrade the switch, ISSU will automatically change the SPB-MGMT route preference to the next lowest value below 150 if there is a conflict with another protocol previously configured with the same route preference value. After an ISSU upgrade, use the **write memory** command to ensure any changes are retained.

Sample Command Configuration

This section provides the sequence of commands used on each switch to configure in-band management for the example shown in "SPB In-Band Management - Route Redistribution" on page 7-26.

BEB-1:

```
-> system name BEB-1
-> spb bvlan 4001
-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001
-> spb interface port 1/1/3
-> spb isis admin-state enable
-> ip interface bvlan-intf address 100.0.0.1 vlan 4001
-> ip interface Loopback0 10.0.0.1
-> ip static-route 128.251.1.0/24 gateway 20.0.0.254
-> ip route-map spb-mgmt-route sequence-number 10 action permit
-> ip route-map spb-mgmt-route sequence-number 10 match ip-address 128.251.1.0/24
redist-control no-subnets
-> ip redist static into spb-mgmt route-map spb-mgmt-route
-> ip route-pref spb-mgmt 155
```

BEB-2:

```
-> system name BEB-2
-> spb bvlan 4001
-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001
-> spb interface port 1/1/3
-> spb isis admin-state enable
-> ip interface bvlan-intf address 100.0.0.2 vlan 4001
-> ip interface Loopback0 10.0.0.2
-> ip redist ospf into spb-mgmt all-routes
-> ip redist spb-mgmt into ospf all-routes
```

BEB-3:

```
-> system name BEB-3
-> spb bvlan 4001
-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001
-> spb interface port 1/1/3
-> spb isis admin-state enable
-> ip interface bvlan-intf address 100.0.0.3 vlan 4001
-> ip interface Loopback0 10.0.0.3
-> ip static-route 128.251.2.0/24 gateway 20.0.0.253
-> ip route-map spb-mgmt-route sequence-number 10 action permit
-> ip route-map spb-mgmt-route sequence-number 10 match ip-address 128.251.2.0/24
redist-control no-subnets
-> ip redist static into spb-mgmt route-map spb-mgmt-route
-> ip route-pref spb-mgmt 155
```

BEB-4:

```
-> system name BEB-4
-> spb bvlan 4001
```

```
-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001
-> spb interface port 1/1/3
-> spb isis admin-state enable
-> ip interface bvlan-intf address 100.0.0.4 vlan 4001
-> ip interface Loopback0 10.0.0.4
```

BCB-1:

```
-> system name BCB-1
-> spb bvlan 4001
-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001
-> spb interface port 1/1/3
-> spb isis admin-state enable
-> ip interface bvlan-intf address 100.0.0.5 vlan 4001
-> ip interface Loopback0 10.0.0.5
```

BCB-2:

```
-> system name BCB-2
-> spb bvlan 4001
-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001
-> spb interface port 1/1/3
-> spb isis admin-state enable
-> ip interface bvlan-intf address 100.0.0.6 vlan 4001
-> ip interface Loopback0 10.0.0.6
```

Verifying the In-Band Management Configuration

Configuring a single IPv4 interface for an SPB Control BVLAN provides in-band management access to the SPB domain. If the underlying configuration is not valid (more than one IP interface, the Control BVLAN is not operational), the **show ip interface** command will display the “Operational State Reason” as “invalid-cfg”. For example:

```
-> show ip interface bvlan-intf
Interface Name = spb-mgmt
  SNMP Interface Index           =   13600003,
  IP Address                     =   11.0.0.47,
  Subnet Mask                    =   255.0.0.0,
  Broadcast Address              =   11.255.255.255,
  Device                         =   vlan 4000,
  Encapsulation                  =   eth2,
  Forwarding                     =   disabled,
  Administrative State           =   enabled,
  Operational State              =   down,
  Operational State Reason       =   invalid-cfg,
  Maximum Transfer Unit          =   1524,
  ARP Count                      =   0,
  Router MAC                     =   e8:e7:32:a4:63:21,
  Local Proxy ARP                =   disabled,
  Primary (config/actual)        =   no/yes
```

ISIS-SPB generates static ARP entries that map the system MAC address to the Control BVLAN IP address of the advertising manager BEB. The **show arp** command displays these entries with the ARP flag set to "M" (Managed). For example:

```
-> show arp

Total 2 arp entries
 Flags (P=Proxy, A=Authentication, V=VRRP, B=BFD, H=HAVLAN, I=INTF, M=Managed)

 IP Addr           Hardware Addr     Type    Flags  Port         Interface   Name
----------------+------------------+-------+-----+-------------+-----------+-------
 100.0.0.1       e8:e7:32:1e:4c:88  STATIC     M          1/1/3  v4001
 100.0.0.3       e2:e7:32:1e:4b:f8  STATIC     M          1/1/3  v4001
```

To verify the SPB-MGMT route redistribution configuration, use the **show ip redist** command. For example:

```
-> show ip redist

Source       Destination
Protocol     Protocol     Admin State  Route Map
------------+------------+-----------+--------------------
LOCAL4       OSPF         Enabled     auto-configure
STATIC4      SPB-MGMT     Enabled     spb-mgmt-route
OSPF         SPB-MGMT     Enabled     none (all-routes)
SPB-MGMT     OSPF         Enabled     none (all-routes)
```

To verify the route preference assigned to SPB-MGMT routes (default is 150), use the **show ip route-pref** command. For example:

```
-> show ip route-pref
  Protocol    Route Preference Value
------------+------------------------
  Local                 1
  Static                2
  OSPF                110
  ISISL1              115
  ISISL2              118
  RIP                 120
  SPB-MGMT            150
  EBGP                190
  IBGP                200
  Import              210
```

SPB-MGMT route information is also included in the **show ip router database** and **show ip routes** commands. For example:

```
-> show ip router database protocol spb-mgmt
Legend: + indicates routes in-use
        b indicates BFD-enabled static route
        i indicates interface static route
        r indicates recursive static route, with following address in brackets

Destination      Gateway       Interface   Protocol   Metric  Tag   Misc-Info
----------------+-------------+-----------+----------+-------+-----+-----------
+ 128.251.2.0/24  20.0.0.253   bvlan-intf   SPB-MGMT   11      0

-> show ip routes

+ = Equal cost multipath routes
 Total 7 routes
```

```
  Dest Address        Gateway Addr        Age        Protocol
-----------------+-------------------+----------+-----------
  1.1.1.1/32          1.1.1.1           4d 5h      LOCAL
  128.251.2.0/24      20.0.0.253        00:10:52   SPB-MGMT
  20.0.0.0/8          20.0.0.1          4d 5h      LOCAL
  24.0.0.0/8          0.0.0.0           4d 5h      STATIC
  40.0.0.0/8          40.0.0.1          4d 5h      LOCAL
  44.0.0.0/8          20.44.44.44       4d 5h      STATIC
  127.0.0.1/32        127.0.0.1         4d 5h      LOCAL
```

The learned SPB-MGMT routes are advertised in an SPB LSP under the TLV list heading “IPv4 Reach Routes”. Use the **show spb isis database** command to display the LSP information. For example:

```
-> show spb isis database lsp-id 00d0.9501.084c.00-00
Legends : P    = The Partition repair bit is set
          OV   = The overload bit is set
          ATT  = The Attach bit is set
          L1   = Specifies a Level 1 IS type
          L2   = Specifies a Level 2 IS type
SPB ISIS LSP Database:
--------------------------------------------------------------------------------
LSP ID        : 00d0.9501.084c.00-00                    Level     : L1
Sequence      : 0x04             Checksum  : 0xc7f1     Lifetime  : 763
Version       : 1                Pkt Type  : 18         Pkt Ver   : 1
Attributes    : L1               Max Area  : 3
SysID Len     : 6                Used Len  : 145        Alloc Len : 1492

TLVs :
Area Addresses          :
   Area Address         : (01) 00
   Area Address         : (03) 00.00.00
Supp Protocols          :
   Protocols            : SPB
IPv4 I/F Addresses      :
   IPv4 Address         : 100.0.0.1
IPv4 Reach Routes       :
   SPB-MGMT route       : 10.0.0.1/32
   SPB-MGMT route       : 128.251.1.0/24
IS-Hostname             :
   Hostname             : DUT1
TE IS Neighbors         :
   Neighbor             : 00d0.9501.b89c SPB Metric 10 Num of Ports 1 Port-Id
0x120(1/3/32)
MT Capability           :
   MT-ID : 0x0
   SPB INSTANCE         :
     CIST Root-ID: 0x0 0x0
     CIST Ext Root Path Cost: 0x00000000 Bridge Priority: 0x8000
     SPSourceID: 0x0011084c (Auto) Number of Trees: 2
     [#1 ] ECT-algo:0x0080c201 baseVid: 4001 spVid: 0 usedByISID: 1(I-SID) mode:
1(SPBM)
     [#2 ] ECT-algo:0x0080c202 baseVid: 4002 spVid: 0 usedByISID: 0() mode: 1(SPBM)
MT Capability :
   MT-ID : 0x0
   SPB SVCID-UCAST-ADDR :
     B-MAC 00.d0.95.01.08.4c Base-VID 4001
     [ISID# 1] 16776961 (T=1/R=1)
```

Interaction With Other Features

This section contains important information about Shortest Path Bridging MAC (SPBM) interaction with other OmniSwitch features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

Backbone VLANs (VLAN Manager)

VLAN Manager CLI commands are used to create an SPB backbone VLAN (BVLAN). Although a BVLAN is created in a similar manner as a standard VLAN, BVLANs differ from standard VLANs as follows:

- No Spanning Tree control—the Spanning Tree protocol is automatically disabled on each BVLAN and all ports associated with each BVLAN will remain in a forwarding state. However, Spanning Tree can remain operational on other types of VLANs.
- No source MAC address learning—normal hardware learning is disabled on BVLANs. Instead, the forwarding database (FDB) is populated by the ISIS-SPB protocol.
- There is no flooding of unknown destination or multicast frames.
- Ingress filtering based on the source MAC address—frames received on ports that do not have an incoming source MAC address pre-programmed by ISIS-SPB are discarded.
- IP interfaces are not supported on BVLANs, except for the Control BVLAN to provide in-band management support for the SPB domain.

Automatic Fabric BVLANs

In previous releases, Automatic Fabric created 16 BVLANs during the SPB discovery phase. The current implementation creates 4 BVLANs. When upgrading from a previous release or using an SPB network with some switches running the current release and others running a previous release, pruning unused BVLANs is recommended to improve SPB scalability and convergence time. Refer to "SPB Network Scalability and Convergence" on page 7-41 for more information.

IP Multicast Switching

In a networking environment where IP multicast traffic is used, destination hosts signal their intent to receive a specific IP multicast stream by sending an Internet Group Management Protocol (IGMP) request to a nearby switch. This process is referred to as IGMP Snooping. The switch then learns on which ports multicast group subscribers are attached and can intelligently deliver traffic only to the respective ports. The OmniSwitch implementation of IGMP Snooping is called IP Multicast Switching (IPMS).

IGMP Snooping for SPB services is essentially the same. An SPB Backbone Edge Bridge (BEB) will apply the logic of IGMP Snooping on a per-service basis to limit the traffic going out of each Service Access Point (SAP) port, as well as limit traffic going out across each backbone port. The SPB bridge will monitor the IGMP queries and requests from SAPs and Service Distribution Point (SDP) ports (also referred to as network virtual ports) to build the stream membership association logic and timing in the same manner as is done on a standard IGMP Snooping bridge.

When traffic arrives on a SAP port, the switch will examine the packet to see if there are any known receivers. If there are any such receivers, then only ports (including network virtual ports) will have a copy of that frame sent on them. When traffic arrives from the core on a network virtual port, the same logic is applied so that a copy of the frame is only sent out on a port where a listener has requested

membership to the stream. However, traffic from the core is never sent back into the core (split horizon protection).

IPMS is configurable in both the VLAN and service domains. Enabling IPMS functionality specifically for SPB services is required to activate IGMP Snooping in an SPB network. See the "Configuring IP Multicast Switching" chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide* for more information.

Link Aggregation

- Both static and dynamic link aggregates are configurable as SPBM service access ports and as SPBM network interfaces.
- Note that a link aggregate must consist of all access ports or all network ports. SPBM functionality is not supported on link aggregates that consist of a mixture of SPBM ports and standard switch ports.
- When creating a link aggregate that will serve as an SPBM service access port or network interface, specify the Tunnel Protocol hashing option for the aggregate. This will ensure that hashing is done on the payload of encapsulated SPB packets. See the "Link Aggregation Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information.

OAM

- OAM support per the IEEE 802.1ah standard for Provider Backbone Bridging (PBB) is applied at the customer VLAN (CVLAN) and the backbone VLAN (BVLAN) level. Support at the service instance (I-SID) level is provided through Remote Fault Propagation on SPB.
- The OmniSwitch Ethernet OAM feature is required to support RFP for SPB. When an RFP local end point is created on the switch, the following reserved maintenance domain and maintenance association is created:
  - RFP_OVER_SPB_DOMAIN_LEVEL*x* (where *x* is the level number specified when a local RFP end point is created)
  - RFP_OVER_SPB_ASSOCIATION
- In addition, the OmniSwitch proprietary Layer 2 ping and traceroute features are available to troubleshoot CVLAN and BVLAN domains, including an I-SID check.

Quality of Service (QoS)

- The priority assignment of a user frame is determined at an access point. A Service Access Point (SAP) on an SPB access port can be configured as trusted or un-trusted. If a SAP is configured as trusted, then internal priority for ingress traffic on that SAP is derived from tagged or NULL tagged ingress packet priority or from default port priority if ingress packet is untagged. If a SAP is untrusted then internal priority can be configured by the user.
- QoS performs the following actions on ports configured as access ports:
  - Access ports are automatically trusted and the default classification is set to 802.1p.
  - The trust status and classification are not user-configurable on access ports.
  - All QoS CLI configuration is blocked on access ports.This includes physical ports and ports that are members of a link aggregate.

- Untagged L2 control packets (such as BPDU, GVRP, AMAP) are tunneled (if enabled) through the SPB domain with the priority value set to 0. Trusted and untrusted SAPs configured on access ports will not affect the priority assignment for Layer 2 control packets.

- QoS priority (802.1p) is applied as follows to trusted and untrusted SAPs:

SAP Configuration	Allowed Configuration	
Tagged (VLAN 1–4094)	**Trusted**	**Tagged traffic priority derived from tags.**
	Untrusted	**Tagged traffic priority configured by user.**
QinQ (outer VLAN 1–4094)	**Trusted**	**Tagged traffic priority derived from outer tags.**
	Untrusted	**Tagged traffic priority configured by user.**
Wild Card	**Trusted**	**Tagged traffic priority derived from tags. Untagged traffic Port default (PRI 0).**
	Untrusted	**Tagged/ traffic priority configured by user**
Untagged	**Trusted**	**Untagged Traffic Port default (PRI 0)**
	Untrusted	**Priority configured by user.**

  - By default, a SAP is trusted with best effort priority (0).
  - A SAP can be dynamically changed to trusted/untrusted without administratively taking down the SAP.
  - A SAP priority may only be set when a SAP is untrusted.
  - When a SAP is changed from untrusted to trusted, any previously assigned priority is reset with best effort priority (0).
  - A trusted SAP that defines a double-tagged encapsulation (QinQ) will use the outer VLAN tag to determine the priority of the frame.

- Priority handling at the edge and core components of an SPBM topology:
  - On a ingress Backbone Edge Bridge (BEB), a frame is classified to a SAP. The internal priority is determined based on the QoS settings of the SAP (for example, trusted vs. untrusted, default priority). This internal priority is mapped to the backbone VLAN (BVLAN) tag of the tunnel encapsulation.
  - The Backbone Core Bridge (BCB) acts as a Layer 2 device that switches the frame across ingress to egress ports in the BVLAN domain. The BVLAN tag is used to determine the internal priority queue on the egress port where the frame is enqueued.
  - On an egress BEB, the internal priority is determined from the BVLAN tag. The frame is de-encapsulated and enqueued to the egress queue(s) of the access port(s) based on this internal priority.

Universal Network Profiles (UNP)

Integration with Virtual Machine Network Profiles (vNPs) to support device discovery and mobility. The UNP feature supports two types of profiles: VLAN and service. A service profile can be configured to classify traffic for SPB or VXLAN tunneling.

The OmniSwitch supports both a VLAN and service domain for traffic classification.

- The VLAN domain is identified by a VLAN ID. In the VLAN domain, each VLAN is accessed through a physical port. Each physical port can have more than one VLAN attached. UNP VLAN classification associates a MAC address to a specific VLAN on a physical UNP bridge port.

- The service domain is identified by one of the following:
  - A Shortest Path Bridging (SPB) service instance identifier (I-SID), which is associated with a Service Manger service ID to represent a virtual forwarding instance (VFI).
  - A VXLAN Network Identifier (VNI), which is associated with a Service Manager service ID to represent a VFI.

In the service domain, each VFI is accessed through a virtual port, referred to as a Service Access Point (SAP). UNP service classification associates a device MAC address to a SAP.

Dynamic Service Access Points

A UNP service profile can trigger the dynamic creation of a SAP when traffic received on a UNP access port is classified and assigned to that profile. If the service (SPB or VXLAN) that the SAP is associated with does not exist, the service is also dynamically created.

Allowing incoming traffic to trigger dynamic SAP creation reduces the amount of manual configuration required. In addition, no other protocols are required on the switch or host device to support this functionality.

UniDirectional Link Detection (UDLD)

UDLD protocol control frames (destination MAC address is 01:00:0c:cc:cc:cc) are processed as follows:

UDLD Status	User Access Port	Network Port (Tagged)	Network Port (Untagged)	Legacy
Globally disabled	tunnel	tunnel	discard	tunnel
Globally enabled	tunnel	tunnel	discard	drop
Enabled on port	peer	tunnel	peer	peer

VRF

IP over SPB uses Virtual Routing and Forwarding (VRF) instances to exchange routes with I-SIDs. This is accomplished via the Global Route Manager (GRM). VRF routes are exported to the GRM table and imported into I-SIDs; I-SID routes are exported to the GRM table and imported into VRFs.

- A binding is created between a VRF and the I-SIDs to identify which I-SIDs will export routes to the GRM for the specified VRF (see "Configuring IP over SPB" on page 7-73 for more information).
- The **ip import** command has an optional **isid** parameter that notifies GRM to import the routes from the specified I-SID into the requesting VRF.

Quick Steps for Configuring SPBM

This section provides a quick tutorial for configuring the SPBM network backbone (control plane) and the service encapsulation path (data plane). The Command Line Interface (CLI) commands provided in this section are used to configure the "Sample SPBM Network Topology" on page 7-12.

Quick Steps for Configuring the SPBM Backbone

The following quick steps are used on each switch in the SPBM backbone that will participate in the "Sample SPBM Network Topology" on page 7-12. This includes both edge and transit (core) switches.

1 Use the **system name** command to assign a unique system name to each SPB switch in the domain.

```
-> system name BEB-1
-> system name BEB-2
-> system name BCB-1
-> system name BCB-2
-> system name BCB-3
-> system name BCB-4
-> system name BCB-5
-> system name BCB-6
```

2 Use the **spb bvlan** command to create BVLANs 4001 and 4002 on each switch (edge and core switches) that will participate in the SPBM topology.

```
-> spb bvlan 4001
-> spb bvlan 4002
```

3 Use the **spb isis bvlan ect-id** command to change the equal cost tree (ECT) algorithm ID for the specified BVLAN, if necessary, to make sure that the same ECT ID is assigned to the same BVLAN ID on each switch (edge and core switches) in the SPBM topology.

```
-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2
```

4 Use the **spb isis control-bvlan** command to designate one of the BVLANs on each SPB switch (edge and core switches) as the Control BVLAN for the SPB instance. The Control BVLAN is used to exchange ISIS-SPB control packets with neighboring SPB switches.

```
-> spb isis control-bvlan 4001
```

5 Use the **spb isis interface** command to configure a port or link aggregate as an SPB interface. This type of interface sends PDUs to detect neighboring SPB switches and form adjacencies and also serves as a network port that is used to carry encapsulated service traffic through the SPBM backbone network.

```
-> spb isis interface port 1/1-4
-> spb isis interface port 1/1-3
```

In the "Sample SPBM Network Topology" on page 7-12, ports 1/1-4 are configured as SPB interfaces on the core switches (BCB-1, BCB-2) and ports 1/1-3 are configured as SPB interfaces on all other switches.

6 Use the **spb isis admin-state** command to enable the SPB instance for the switch. Enabling ISIS-SPB on the switch triggers the transmission of hello packets from the SPB interfaces, which starts the process of defining the SPB infrastructure and calculating the shortest path trees (SPTs) through the topology.

```
-> spb isis admin-state enable
```

Quick Steps for Configuring SPB Services

The following quick steps use the OmniSwitch Service Manager commands to configure the logical entities that comprise the SPB services in the "Sample SPBM Network Topology" on page 7-12.

1 Use the **service access** command to configure a port or link aggregate on which customer traffic is received as an SPB service access port.

```
-> service access port 1/12
-> service access port 2/1
```

2 Use the **service spb** command to create an SPB service and associate that service with a backbone service instance identifier (I-SID) and BVLAN.

```
-> service 1 spb isid 500 bvlan 4001 admin-state enable
-> service 2 spb isid 501 bvlan 4002 admin-state enable
```

3 Use the **service sap** command to create a service access point (SAP) by associating an SPB service with SAP ID. A SAP ID is comprised of a port or link aggregate and an encapsulation value that identifies the customer traffic to associate with the service.

```
-> service 1 sap port 1/12:10 admin-state enable
-> service 1 sap port 2/1:10 admin-state enable
-> service 2 sap port 1/12:0 admin-state enable
-> service 2 sap port 2/1:all admin-state enable
```

In this example,

- SPB service 1 (I-SID= 500, BVLAN=4001) is assigned to SAPs 1/12:10 and 2/1:10. Traffic received on the SAP access ports (1/12, 2/1) that has an outer tag (customer VLAN tag) equal to 10 is mapped to SPB service 1.
- SPB service 2 (I-SID=501, BVLAN=4002) is assigned to SAPs 1/12:0 and 2/1:all. All tagged traffic (except for VLAN 10 tagged traffic, which is mapped to service 1) and untagged traffic received on the SAP access ports (1/12, 2/1) is mapped to SPB service 2.

Sample Command Configuration

This section provides the sequence of commands used on each switch to configure the "Sample SPBM Network Topology" on page 7-12. Note that the SPBM backbone is configured on every switch first, then the SPBM service architecture is configured second. Following this order of configuration is highly recommended to ensure proper switch participation in ISIS-SPB adjacencies and shortest path tree calculations.

SPBM Backbone Commands

The **system name** and Shortest Path Bridging (**spb**) commands are used to configure the SPBM backbone infrastructure for the sample topology, as shown:

BEB-1	**BEB-2**	**BCB-1**
-> system name BEB-1	-> system name BEB-2	-> system name BCB-1
-> spb bvlan 4001	-> spb bvlan 4001	-> spb bvlan 4001
-> spb bvlan 4002	-> spb bvlan 4002	-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1	-> spb isis bvlan 4001 ect-id 1	-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2	-> spb isis bvlan 4002 ect-id 2	-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001	-> spb isis control-bvlan 4001	-> spb isis control-bvlan 4001
-> spb interface port 1/1-3	-> spb interface port 1/1-3	-> spb interface port 1/1-4
-> spb isis admin-state enable	-> spb isis admin-state enable	-> spb isis admin-state enable

BCB-2	BCB-3	BCB-4
-> system name BCB-2	-> system name BCB-3	-> system name BCB-4
-> spb bvlan 4001	-> spb bvlan 4001	-> spb bvlan 4001
-> spb bvlan 4002	-> spb bvlan 4002	-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1	-> spb isis bvlan 4001 ect-id 1	-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2	-> spb isis bvlan 4002 ect-id 2	-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001	-> spb isis control-bvlan 4001	-> spb isis control-bvlan 4001
-> spb interface port 1/1-4	-> spb interface port 1/1-3	-> spb interface port 1/1-3
-> spb isis admin-state enable	-> spb isis admin-state enable	-> spb isis admin-state enable

BCB-5	BCB-6
-> system name BCB-5	-> system name BCB-6
-> spb bvlan 4001	-> spb bvlan 4001
-> spb bvlan 4002	-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1	-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2	-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001	-> spb isis control-bvlan 4001
-> spb interface port 1/1-3	-> spb interface port 1/1-3
-> spb isis admin-state enable	-> spb isis admin-state enable

SPBM Service Commands

The Service Manager (**service**) commands are used to build the SPBM services architecture for the sample topology, as shown. Note that services are only configured on designated BEB switches.

BEB-1	BEB-2
-> service access port 1/12	-> service access port 2/1
-> service 1 spb isid 500 bvlan 4001	-> service 1 spb isid 500 bvlan 4001
-> service 2 spb isid 501 bvlan 4002	-> service 2 spb isid 501 bvlan 4002
-> service 1 sap port 1/12:10 admin-state enable	-> service 1 sap 2/1:10 admin-state enable
-> service 2 sap port 1/12:0 admin-state enable	-> service 2 sap 2/1:0 admin-state enable
-> service 2 sap port 1/12:all admin-state enable	-> service 2 sap 2/1:all admin-state enable

Configuring SPBM

Configuring the SPBM backbone and service layers requires several steps. These steps are outlined here and further described throughout this section. For a brief tutorial on configuring SPBM, see "Quick Steps for Configuring SPBM" on page 7-36.

Configure the SPBM Backbone (ISIS-SPB)

Only switches that are SPB capable can participate in the SPBM network topology. The following configuration steps are required to make an OmniSwitch an SPB-capable node:

1 Create a BVLAN. The BVLAN provides the foundation of the SPBM infrastructure. A BVLAN is associated with an equal cost tree (ECT) algorithm ID and an SPB service instance ID that is used to carry customer traffic through the backbone network. See "Backbone VLANs" on page 7-40.

2 Configure SPB interfaces. An SPB interface is associated with each BVLAN that is configured on the switch. At the ISIS-SPB level, this type of interface sends and receives ISIS Hello packets and link state PDU (LSP) to discover adjacent SPB switches and calculate the shortest path trees through the SPBM network topology. At the services level, the SPB interfaces serve as network ports that are used to carry encapsulated customer traffic through the network. See "Configuring SPB Interfaces" on page 7-44.

3 Configure global ISIS-SPB parameters. In addition to enabling/disabling the ISIS-SPB instance for the switch, global configuration includes settings such as a system name for the switch, global bridge parameters, and various wait time intervals. When ISIS-SPB is enabled for the switch, default settings for these global bridge parameters and wait time intervals are active. It is only necessary to change these values if the default settings are not sufficient. See "Configuring Global ISIS-SPB Parameters" on page 7-47.

For more information about SPBM commands, see Chapter 9, "Shortest Path Bridging Commands," in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configure SPBM Services

The OmniSwitch Service Manager application is used to configure the services layer of the SPBM network topology. A service is defined by a specific set of logical entities that are configured only on the backbone edge bridges (BEBs) of the network. The following configuration steps are required to define a service-based architecture for an SPBM network:

1 Create an SPBM service. A Service Manager service ID is associated with a BVLAN, a backbone service instance identifier (I-SID), and a service access point (SAP) to identify the customer traffic that the service will tunnel through the provider network. See "Creating an SPB Service" on page 7-52.

2 Configure access (customer-facing) ports. One or more access ports are associated with a service access point (SAP) to identify to the service which ports will receive customer traffic that the service will process for tunneling through the provider network. When an access port is associated with a SAP, the SAP parameter attributes are applied to traffic received on the access port. See "Configuring Service Access Ports" on page 7-56.

3 Define access port profile attributes. A default Layer 2 profile is automatically assigned to an access port at the time the port is configured as an access port. This profile determines how control frames received on the port are processed. It is only necessary to configure a Layer 2 profile if the default attribute values are not sufficient. See "Configuring Layer 2 Profiles for Access Ports" on page 7-57.

4 Configure an SPB service access point (SAP). A SAP binds an SPB service to an access (customer-facing) port and defines which customer traffic to tunnel through the service. Each SAP is associated to one service name, but a single service can have multiple SAPs to which it is associated. See "Configuring Service Access Points (SAPs)" on page 7-55.

To define a Remote Fault Propagation (RFP) domain to monitor SPB services, see "Configuring Remote Fault Propagation for SPBM" on page 7-66.

For more information about Service Manager commands, see Chapter 10, "Service Manager Commands," in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

SPB Configuration Guidelines

Configuring an SPBM network topology involves setting up two layers of functionality: the ISIS-SPB backbone infrastructure and the Provider Backbone Bridge (802.1ah) services layer for MAC-in-MAC encapsulation. Review the guidelines in this section before attempting to configure the various components of the SPBM infrastructure and services.

ISIS-SPB

This implementation of the ISIS-SPB protocol supports only a single topology with a multi-topology identifier (MT-ID) of zero.

- The ISIS-SPB protocol instance is independent of IP IS-IS, or other network layer protocol identifiers (NLPIDs) riding in the same IS-IS implementation. However, ISIS-SPB and IP IS-IS can coexist on the same switch.
- ISIS-SPB interfaces, link state packet databases (LSPDB), and forwarding information are all created and maintained within the single ISIS-SPB instance.
- IS-IS Level 1 point-to-point adjacencies are supported; Level 2 is not supported at this time.
- SPB interfaces are associated with a link metric cost that is configurable, thus providing the ability to change the logical topology created by the ISIS-SPB instance. However, if different metric values are configured on each side of a link, ISIS-SPB will choose the higher-valued one as the metric to use for both sides. This is necessary to enforce the symmetry of SPT calculations in both directions across the link.
- Enabling SPB for the switch automatically triggers the transmission of Hello packets from the SPB interfaces, thus starting the process of discovery and forming adjacencies to build shortest path trees.

Backbone VLANs

- The backbone VLAN (BVLAN) configuration must be the same on each SPB switch within the PBB network.For example, if BVLAN 10 with an ECT ID of 1 is configured on one switch, then BVLAN 10 with an ECT ID of 1 must exist on all other SPB bridges in the network to ensure proper calculation of the ISIS-SPB shortest path trees through the backbone.
- In most cases, one BVLAN is sufficient for virtualizing traffic through the network backbone. However, configuring more than one BVLAN provides alternate routes for tunneling customer traffic. This can also provide a form of load balancing by distributing traffic over different BVLAN segments.
- If more than one BVLAN is needed, configure each BVLAN with a different ECT algorithm ID. For example, if two BVLANs (BVLAN 4001 and BVLAN 4002) are needed for a specific SPBM

topology, then create BVLAN 4001 with ECT ID 1 and BVLAN 4002 with ECT ID 2 on each switch that is going to participate in the topology.

Notes:

- When adding another BVLAN to an existing SPBM topology instance, create the new BVLAN and its associated ECT ID on every switch first, then configure the SPB service association for the BVLAN. Creating SPB services before the BVLAN configuration is complete on all switches can cause problems with forming adjacencies or may even cause an SPB switch to drop existing adjacencies.
- Limiting the number of BVLANs to a maximum of four is highly recommended to improve SPB network scalability and convergence time.

- All encapsulated traffic within the BVLAN domain is unicast with resolved source and destination BMAC addresses. Frames received on BVLAN ports that do not have an incoming source MAC address pre-programmed by ISIS-SPB are discarded.

SPB Network Scalability and Convergence

Typically, the number of BVLANs configured is determined by the number of ECT paths required. Although multiple BVLANs can be created, it is highly recommended to limit the number BVLANs to a maximum of four. Doing so reduces the scale of address updates across the control plane and improves network scalability, stability, and convergence.

If there are more than four BVLANs configured in the SPB network, the number of BVLANs can be reduced by removing unused BVLANs and, if necessary, consolidating SPB services that are distributed across more than four BVLANs.

Removing Unused BVLANs

Removing unused BVLANs is not a local task that is done on just one specific switch, as a BVLAN is configured on all participating SPB switches in the network. To remove a BVLAN, first check to see if the BVLAN is idle or active. This is done by checking to see if the BVLAN is attached to an SPB service.

If the BVLAN is attached to a service on the local switch or through any other switch in the network, it is considered active and cannot be deleted. The service attachment does not have to be to the local BVLAN ID. It can be to the same BVLAN ID on any other SPB switch.

To check whether a BVLAN is active or idle, use the **show spb isis bvlans** command. For example:

```
-> show spb isis bvlans
SPB ISIS BVLANS:
                                    Services  Num    Tandem     Root Bridge
 BVLAN   ECT-algorithm     In Use   mapped    ISIDS  Multicast  (Name : MAC Address)
-------+-----------------+-------+---------+------+----------+--------------------
  4000  00-80-c2-01       YES     YES           1  SGMODE
  4001  00-80-c2-02       YES     YES           1  SGMODE
  4002  00-80-c2-03       NO      NO            0  SGMODE
  4003  00-80-c2-04       NO      NO            0  SGMODE

BVLANs:    4
```

If a “YES” appears in the “In Use” column, a service is attached to the BVLAN ID. Because this is a network-wide view, the BVLAN is active even if the service is not configured locally on the switch. A remote switch may have a service attached to the same BVLAN ID, while the local switch serves as a transit switch for the active BVLAN.

If a “NO” appears in the “In Use” column, the BVLAN is idle and can be deleted without having to remove any associated SPB service objects. In the above **show spb isis bvlans** example, BVLANs 4002

and 4003 are idle. To delete one or both of these BVLANs, use the **no** form of the **spb bvlan** command. For example:

```
-> no spb bvlan 4002
-> no spb bvlan 4003
```

Removing idle BVLANs will not have any effect on the SPB network.

Consolidating SPB Services

If SPB services are distributed across more than four BVLANs, consolidating the services across four or less BVLANs is recommended. This process involves deleting active BVLANs, which cannot be done without first removing the service objects (service IDs, SAPs) that are associated with the BVLAN.

Note. The SPB service association with a BVLAN cannot be changed on the fly, so the service ID and SAP have to be completely removed. As a result, there will be service down time until the service ID and SAP are configured again on a different BVLAN.

Refer to "Configure SPBM Services" on page 7-39 for more information about creating and removing SPB service objects.

Configuring BVLANs

The SPBM backbone VLAN (BVLAN) provides the foundation on which ISIS-SPB shortest path trees are built and SPBM services tunnel encapsulated customer data through the Provider Backbone Bridge network (PBBN). Configuring a BVLAN on a switch is also the first step in setting up the ISIS-SPB infrastructure and in making an OmniSwitch an SPB-capable node.

Note. The BVLAN configuration must be the same on each OmniSwitch that is going to participate in the SPBM network topology. So if BVLAN 4001 is created on one switch, then BVLAN 4001 must be created on all other switches in the SPBM network.

To create a BVLAN, use the **spb bvlan** command with the optional **name** parameter. For example:

```
-> spb bvlan 4001 name spb-4001
```

If the **name** parameter is not specified with this command, the VLAN ID is used for the name by default. For example, the following command creates BVLAN 4001 with "VLAN 4001" as the name:

```
-> spb bvlan 4001
```

To remove a BVLAN, use the **no** form of the **spb bvlan** command. For example:

```
-> no spb bvlan 4001
```

Assigning the Equal Cost Tree ID

ISIS-SPB calculations may result in multiple paths of equal costs. The Equal Cost Tree (ECT) ID specifies a tie-breaking algorithm that is used when ISIS-SPB is calculating a set of shortest path trees from one switch to all other switches in the SPB domain. When a BVLAN is created, an ECT ID is automatically assigned to the BVLAN. If it is the first BVLAN created on the switch, ECT ID 1 is assigned, otherwise the next available ID number is used.

Each BVLAN created must be duplicated on all other participating switches in the SPBM network and must use the same ECT ID number for that BVLAN on each switch. A BVLAN created on one switch

may not be automatically assigned the same ECT ID on another switch. As a result, it may be necessary to modify the ECT ID number using the **spb isis bvlan ect-id** command. For example:

```
-> spb isis bvlan 4002 ect-id 2
```

Note. When adding another BVLAN to an existing SPBM topology instance, create the new BVLAN and its associated ECT ID on every switch first, then configure the SPB service association for the BVLAN. Creating SPB services before the BVLAN configuration is complete on all switches can cause problems with forming adjacencies or may even cause an SPB switch to drop existing adjacencies.

Configuring the Control BVLAN

One of the BVLANs configured on each switch serves as the Control BVLAN for the ISIS-SPB instance. The Control BVLAN exchanges ISIS-SPB control packets with neighboring SPB switches on behalf of all BVLANs configured on the local switch. The control packets are tagged with the Control BVLAN ID.

To designate a BVLAN as the Control BVLAN, use the **spb isis control-bvlan** command. For example:

```
-> spb isis control-bvlan 4002
```

A Control BVLAN also carries regular encapsulated SPB domain traffic in addition to ISIS-SPB control packets. In other words, a VLAN can serve as both a regular BVLAN and a Control BVLAN at the same time.

Configuring an IP Interface on the Control BVLAN

To configure an IP interface on the Control BVLAN to support in-band management access in the SPBM domain, use the **ip interface** command. (See "SPB In-Band Management" on page 7-25 for more information.)

In the following example, an IP interface is configured for Control BVLAN 4002:

```
-> ip interface "spb-mgmt" address 10.1.1.1/24 vlan 4002
```

Only one Control BVLAN can be configured on a switch, and only IPv4 interface is supported. ISIS-SPB is the only protocol supported in the IP BVLAN domain for exchanging or advertising IP routing information. No other routing protocol (including VRRP) is supported.

Configuring the Tandem Multicast Mode

The tandem multicast mode (*,G) or (S,G) of a BVLAN is applied only to SPB services associated with the BVLAN that are using tandem replication for multicast traffic. When a BVLAN is created, the (S,G) tandem multicast mode is applied by default.

To change the tandem multicast mode for a BVLAN, use the **spb isis bvlan tandem-multicast-mode** command and specify either **gmode** (*,G) or **sgmode** (S,G). For example:

```
-> spb isis bvlan 4001 tandem-multicast-mode sgmode
-> spb isis bvlan 4002 tandem-multicast-mode gmode
```

Verifying the BVLAN Configuration

To view the BVLAN configuration for the switch, use the **show spb isis bvlans** command. For example:

```
-> show spb isis bvlans
SPB ISIS BVLANS:
                                   Services  Num    Tandem     Root Bridge
 BVLAN   ECT-algorithm     In Use  mapped    ISIDS  Multicast  (Name : MAC Address)
-------+-----------------+-------+---------+------+----------+--------------------
  4001  00-80-c2-01       YES     YES          52  SGMODE
  4002  00-80-c2-02       YES     YES          51  GMODE

BVLANs: 2
```

The BVLAN is a special type of VLAN that is created and maintained by VLAN Manager. As a result, it also appears in the VLAN Manager **show** command displays. For example, in the following **show vlan** output display, VLANs 4001 through 4004 are included and “spb” appears in the “type” column:

```
-> show vlan
 vlan  type  admin  oper  ip   mtu   name
-----+------+-----+-----+-----+-----+------------
   1   std    Dis   Dis   Dis  1500   VLAN 1
1000   std    Ena   Ena   Ena  1500   VLAN 1000
4001   spb    Ena   Ena   Dis  1524   VLAN 4001
4002   spb    Ena   Ena   Dis  1524   VLAN 4002
4003   spb    Ena   Ena   Dis  1524   VLAN 4003
4004   spb    Ena   Ena   Dis  1524   VLAN 4004
4094   mcm    Ena   Dis   Dis  9198   MCM IPC
```

To view configuration information for an individual BVLAN, use the **show vlan** command and specify the BVLAN ID. For example:

```
-> show vlan 4001
Name                     : VLAN 4001,
Type                     : Backbone vlan,
Administrative State     : enabled,
Operational State        : disabled,
IP Router Port           : disabled,
IP MTU                   : 1524
```

Configuring SPB Interfaces

A port or link aggregate is configurable as an SPB interface. Each switch in the SPBM topology should have at least one SPB interface configured. The SPB interface serves more than one purpose:

- Advertises IS-IS Hello packets to discover SPB neighbors and establish adjacencies.
- After adjacencies are established, exchanges link state packets (LSPs) with SPB neighbors to build a local LSP database (LSPDB). A switch’s adjacencies are reflected in the contents of its link state packets. This relationship between adjacencies and link state allows the protocol to detect downed routers in a timely fashion.
- Serves as a network port by forwarding encapsulated SPB service traffic on backbone VLANs (BVLANs) through the SPBM Provider Backbone Bridge (PBB) network.

To configure a port or link aggregate as SPB interface, use the **spb isis interface** command. For example:

```
-> spb isis interface port 1/10
-> spb isis interface linkagg 5
```

When a port is converted to an SPB interface, the interface is automatically assigned to all existing BVLANs. There is one ISIS-SPB instance per switch, and each BVLAN and SPB interface is associated with that instance. However, it is also possible to tag SPB interfaces to carry traffic for standard VLANs.

The **spb isis interface** command is also used to optionally configure the following parameter values:

- **admin-state—**Administratively enables or disables the SPB interface. By default, the interface is enabled when the SPB interface is created.
- **hello-interval—**Specifies the amount of time, in seconds, to wait between each transmission of a hello packet from this interface. By default, the hello time interval is set to nine seconds.
- **hello-multiplier—**Specifies an integer value that is multiplied by the hello interval time to determine the amount of time, in seconds, a receiving bridge holds onto the hello packets transmitted from this interface. By default, the hello multiplier is set to three.
- **metric**—An integer value that specifies the link cost to reach the destination backbone MAC (BMAC), By default, the link cost is set to ten. Changing the link metric value provides a method for changing the logical topology as calculated by ISIS-SPB.
- **type**—Specifies whether the SPB interface will operate as a point-to-point (P2P) network interface on which a single adjacency is formed or as a multiple access network interface on which multiple adjacencies are formed. By default, the interface type is set to P2P. Changing the interface type to multiple access facilitates extending an SPB network across a shared network.
- **priority**—Specifies a priority value that is used in a multiple access configuration to determine which interface will serve as a Designated Intermediate System (DIS). This value is only used when the SPB interface type is set to operate as a multiple access network interface. By default, the priority value is set to 64.

The following command examples change the default parameter values for the SPB interface:

```
-> spb isis interface port 1/1/7 hello-interval 60
-> spb isis interface linkagg 3 hello-multiplier 10
-> spb isis interface port 2/1/1 metric 100
-> spb isis interface linkagg 5 hello-interval 20 hello-multiplier 5 metric 200

-> spb isis interface port 2/1/7 type multi-access priority 50
-> spb isis interface port 2/1/7 type p2p
-> spb isis interface linkagg 10 type multi-access priority 100
-> spb isis interface linkagg 10 type p2p
```

Verifying the SPB Interface Configuration

To view the SPB interface configuration for the switch, use the **show spb isis interface** command. For example:

```
-> show spb isis interface
SPB ISIS Interfaces:
                                Oper   Admin   Link     Hello   Hello   Circ
 Interface      Level  CircID  state  state   Metric   Intvl   Mult    Type
--------------+-------+-------+------+-------+-------+-------+-------+------------
 1/1/1          L1      1       DOWN   UP      10      9       3       P2P
 1/1/2          L1      2       UP     UP      10      9       3       P2P
 1/1/3          L1      3       DOWN   UP      10      9       3       P2P
 1/1/4          L1      4       DOWN   UP      10      9       3       P2P
 1/1/5          L1      5       UP     UP      10      3       3       multi-access
 1/1/6          L1      6       DOWN   UP      10      9       3       P2P
 1/1/7          L1      7       DOWN   UP      10      9       3       P2P
 1/1/10         L1      9       DOWN   UP      10      9       3       P2P

Interfaces : 8
```

Configuring Global ISIS-SPB Parameters

This section describes the global configuration for the ISIS-SPB instance, which includes the following:

- "Configuring the System Name" on page 7-48.
- "Configuring the SPB Bridge Priority" on page 7-48.
- "Configuring the ISIS-SPB Area Address" on page 7-48.
- "Configuring the Shortest Path Source ID" on page 7-48.
- "Configuring the Control MAC Address" on page 7-48.
- "Configuring the Shortest Path First Wait Time" on page 7-49.
- "Configuring the Link State Packet Wait Time" on page 7-50.
- "Configuring the Overload State" on page 7-50.
- "Configuring Redundant Switches for Graceful Restart" on page 7-51.
- "Enabling/Disabling ISIS-SPB" on page 7-52.

To verify the global configuration parameter values for the switch, use the **show spb isis info** command. For example:

```
-> show spb isis info
SPB ISIS Bridge Info:
  System Id             = e8e7.3233.1831,
  System Hostname       = BEB-1,
  SPSourceID            = 03-18-31,
  SPBM System Mode      = auto,
  BridgePriority        = 32768 (0x8000),
  MT ID                 = 0,
  Control BVLAN         = 4001,
  Area Address          = 0.0.0,
  Level Capability      = L1,
  Admin State           = UP,
  LSDB Overload         = Disabled,
  Last Enabled          = Thu Aug  2 22:43:19 2012,
  Last SPF              = Fri Aug  3 18:15:51 2012,
  SPF Wait              = Max: 1000 ms, Initial: 100 ms, Second: 300 ms,
  LSP Lifetime          = 1200,
  LSP Wait              = Max: 1000 ms, Initial: 0 ms, Second: 300 ms,
  Graceful Restart      = Disabled,
  GR helper-mode        = Disabled,
  # of L1 LSPs          = 8
  Control Address       = 01:80:C2:00:00:14 (AllL1)
```

Configuring the System Name

Configuring a system name is required on each switch that is going to participate in the SPBM topology. To configure a system name for the switch, use the **system name** command. For example:

```
-> system name BEB-1
```

ISIS-SPB advertises the system name to identify the switch to other SPB peer switches.

Configuring the SPB Bridge Priority

A bridge is ranked within the SPB topology by its bridge ID (an eight byte hex number). The bridge priority value makes up the upper two bytes of the eight-byte SPB bridge ID. The lower six bytes of the Bridge ID contain the system ID, which is the dedicated bridge MAC address of the SPB bridge.

The bridge priority is used in shortest path tree calculations. The lower the priority value, the higher the priority. Setting a different bridge priority value on different SPB bridges will override the system ID significance during the shortest path tree (SPT) calculation.

By default, all SPB switches are assigned a priority value of 32768. To change the bridge priority value for a switch, use the **spb isis bridge-priority** command. For example:

```
-> spb isis bridge-priority 25590
```

Configuring the ISIS-SPB Area Address

By default, the IS-IS area address for the ISIS-SPB instance is set to 0.0.0, which is typically sufficient for this implementation of SPBM. Both ISIS-SPB and ISIS-IP instances may coexist on the same switch as long as they don't use the same area address.

If changing the area address is necessary, use the **spb isis area-address** command. For example:

```
-> spb isis area-address 1.1.1
```

Note. Each switch that is going to participate in the SPB topology must use the same area address and must use an address that is different from the ISIS-IP area address.

Configuring the Shortest Path Source ID

The shortest path (SP) source ID, identifies the source of multicast frames and is relevant only in multicast tandem replication mode. By default, the last three least significant bytes of the system ID (local bridge MAC address) is used for the source ID value.

To change the source ID value, use the **spb isis source-id** command. For example:

```
-> spb isis source-id 07-0b-d3
```

To set the source ID back to the default value, use the **spb isis source-id** command with the **auto** parameter. For example:

```
-> spb isis source-id auto
```

Configuring the Control MAC Address

The control MAC address is the destination MAC address used for ISIS-SPB control packets. Changing this address can enhance interoperability between an SPB-capable OmniSwitch and other third-party SPB-capable devices.

By default, the control MAC address is set to 01:80:C2:00:00:14 (all Level 1 Intermediate Systems). The following parameters are used with the **spb isis control-address** command to change the control MAC address:

- **alll1**—All Level 1 Intermediate Systems (01:80:C2:00:00:14).
- **alll2**—All Level 2 Intermediate Systems (01:80:C2:00:00:15).
- **allis**—All Intermediate Systems (09:00:2B:00:00:05).

For example, the following command changes the default control MAC address from AllL1 to AllL2:

```
-> spb isis control-address alll2
```

Configuring the Shortest Path First Wait Time

The **spb isis spf-wait** command is used to configure the time intervals between the first, second, and subsequent ISIS-SPB shortest path first (SPF) calculations.

Subsequent SPF calculations, if required, are generated at exponentially increasing intervals of the SPF second wait time interval until the maximum wait time interval value is reached. For example, if the second-wait interval value is set to 1000 milliseconds, then the next SPF calculation is triggered after 2000 milliseconds and the next SPF calculation after that is triggered at 4000 milliseconds, and so on, until the maximum-wait interval value is reached.

When the maximum interval value is reached, the SPF wait interval will stay at the maximum value until there are no more SPF calculations scheduled during that interval. After a full interval without any SPF calculations, the SPF wait interval will reset back to the initial wait time interval value.

The following **spb isis spf-wait** command parameters are used to configure the SPF timers:

- **max-wait—**The maximum number of milliseconds to wait between two consecutive SPF calculations. The default maximum wait time value is set to 1000 milliseconds. Specify a maximum value that is the same or greater than the second wait time value.
- **initial-wait—**The number of milliseconds to wait before triggering an initial SPF calculation after a topology change. The default initial wait time value is set to 100 milliseconds. Specify a value that is the same or less than the maximum wait time value.
- **second-wait—**The number of milliseconds to wait between the first and second SPF calculation. The default second wait time value is set to 300 milliseconds. Specify a value that is the same or less than the maximum wait time value.

For example, the following command changes the SPF wait time values for the local SPB instance:

```
-> spb isis spf-wait max-wait 2500 initial-wait 1000 second-wait 1500
```

To change one or more of the wait time values, it is only necessary to specify the parameter for the desired change. For example:

```
-> spb isis spf-wait max-wait 5000
-> spb isis spf-wait initial-wait 1000
-> spb isis spf-wait second-wait 2000
```

To set the wait time values back to the default settings. use the **spb isis spf-wait** command without specifying any of the parameters. For example:

```
-> spb isis spf-wait
```

Configuring the Link State Packet Wait Time

The **spb isis lsp-wait** command is used to configure the time intervals between the first, second, and subsequently generated link state packets (LSPs).

Subsequent LSP, if required, are generated at exponentially increasing intervals of the LSP second wait time interval until the maximum value is reached. For example, if the second-wait interval value is set to 10 seconds, then the next LSP is generation is triggered after 20 seconds and the next LSP generated after that is triggered at 40 seconds, and so on, until the maximum wait time interval value is reached.

When the maximum interval value is reached, the LSP wait interval will stay at the maximum value until there are no more LSP generations during that interval. After a full interval without any LSP generations, the LSP wait interval will reset back to the initial wait time interval value.

The following **spb isis lsp-wait** command parameters are used to configure the SPF timers:

- **max-wait**—The maximum number of seconds to wait between two consecutively generated LSPs. The default maximum wait time value is set to 1000 milliseconds. Specify a maximum value that is the same or greater than the second wait time value.
- **initial-wait**—The number of seconds to wait before triggering an initial LSP generation after a topology change. The default initial wait time value is set to 0 milliseconds. Specify a value that is the same or less than the maximum wait time value.
- **second-wait**—The minimum number of seconds to wait between the first and second generated LSPs. The default second wait time value is set to 300 milliseconds. Specify a value that is the same or less than the maximum wait time value.

For example, the following command changes the LSP wait time values for the local SPB instance:

```
-> spb isis lsp-wait max-wait 2000 initial-wait 1000 second-wait 1500
```

To change one or more of the wait time values, it is only necessary to specify the parameter for the desired change. For example:

```
-> spb isis lsp-wait max-wait 5000
-> spb isis lsp-wait initial-wait 2500
-> spb isis lsp-wait second-wait 3000
```

To set the wait time values back to the default settings. use the **spb isis lsp-wait** command without specifying any of the parameters. For example:

```
-> spb isis lsp-wait
```

Configuring the Overload State

This implementation of ISIS-SPB supports the overload state mechanism, which allows an instance of ISIS-SPB to inform its neighbors that the instance is nearing or exceeding its capabilities. When peers see that a switch is advertising in this state, they will select an alternate path around the overloaded switch.

The ISIS-SPB instance for a switch may dynamically trigger the overload state condition when the instance detects that it is nearing or has reached resource limits. However, it is also possible to manually trigger the overload state condition using the **spb isis overload** command. For example:

```
-> spb isis overload
```

Some advantages of manually triggering the overload state condition, even if the instance is no where near its resource limits, are as follows:

- The switch is designated as “leaf node” that should never carry transit traffic. Configuring the link metric value for the SPB interfaces on the switch and attached peers is another method for preventing the switch from receiving transit traffic, but enabling the overload state is a much quicker way to achieve the same results and requires less configuration.
- When there is a need to remove the switch from service (temporarily or permanently). In this scenario, network availability is increased because peer switches will detect the overload state of the switch and gracefully transition to alternate paths, while the “manually overloaded” switch continues to forward packets. Just simply shutting the switch down would cause more disruption to network traffic.

When the overload state is either dynamically or manually enabled for the switch, the overload bit is set in LSP 0 to indicate that this ISIS-SPB instance is not available to accept transit traffic.

When the overload state is enabled, the switch will operate in this state for an infinite amount of time. To configure the switch to remain in the overload state for only a specific amount of time (in seconds), use the **spb isis overload** command with the optional **timeout** parameter. For example:

```
-> spb isis overload timeout 500
```

To disable the overload state, use the **no** form of the **spb isis overload** command. For example:

```
-> no spb isis overload
```

It is also possible to specify that the overload state is enabled for the switch after every system bootup. This is done using the **spb isis overload-on-boot** command, which also has an optional **timeout** parameter. For example:

```
-> spb isis overload-on-boot timeout 500
```

To disable the overload-on-boot option, use the **no** form of the **spb isis overload-on-boot** command. For example:

```
-> no spb isis overload-on-boot timeout 500
```

Note that the **no spb isis overload** command does not disable the overload-on-bootup option.

Configuring Redundant Switches for Graceful Restart

By default, ISIS-SPB graceful restart is enabled. When graceful restart is enabled, the switch can either be a helper (which helps a neighbor router to restart) or a restarting router, or both. When graceful restart is enabled on the switch, the helper mode is automatically enabled by default.

To configure ISIS-SPB graceful restart support on an OmniSwitch, use the **spb isis graceful-restart** command. For example, to configure graceful restart on the router, enter:

```
-> spb isis graceful-restart
```

The helper mode can be disabled on the switch with the **spb isis graceful-restart helper** command. For example, to disable the helper support for neighboring switches, enter the following:

```
-> ip isis graceful-restart helper disable
```

To disable support for graceful restart, use the **no** form of the **spb isis graceful-restart** command. For example:

```
-> no spb isis graceful-restart
```

Enabling/Disabling ISIS-SPB

By default ISIS-SPB is disabled on the switch. To enable ISIS-SPB, use the **spb isis admin-state** command with the **enable** option. For example:

```
-> spb isis admin-state enable
```

To disable the ISIS-SPB instance on the switch, enter the **spb isis admin-state** command with the **disable** option. When the ISIS-SPB status s disabled for the switch, the related configuration settings and statistics are retained.

```
-> spb isis admin-state disable
```

Note. Enabling ISIS-SPB on a switch starts the process of ISIS-SPB discovery, adjacency building, and shortest path tree calculations. Make sure that the SPBM configuration is set up first, then enable ISIS-SPB on each switch that will participate in the SPBM network.

Creating an SPB Service

An SPB service is identified by a service ID number, which represents an association between a backbone service instance identifier (I-SID) and an existing BVLAN. Basically, creating an SPB service binds the backbone I-SID to a BVLAN ID. All traffic mapped to the specific I-SID is then encapsulated and forwarded on the associated BVLAN to the intended destination.

The **service spb** command is used to create an SPB service. For example, the following command creates SPB service 1 and binds I-SID 100 to BVLAN 4001:

```
-> service 1 spb isid 500 bvlan 4001 admin-state enable
```

The BVLAN ID specified with the **service spb** command must already exist in the switch configuration. However, the I-SID number specified creates a new I-SID that is bound to the BVLAN for this service.

Note. When adding another BVLAN to an existing SPBM topology instance, create the new BVLAN and its associated ECT ID on every switch first, then configure the SPB service association for the BVLAN. Creating SPB services before the BVLAN configuration is complete on all switches can cause problems with forming adjacencies or may even cause an SPB switch to drop existing adjacencies.

Modifying Default SPB Service Parameters

The following SPB service parameter values are set by default at the time the service is created. If necessary, use the specified commands to change the default values.

Parameter Description	Command	Default
Service description.	**service description**	None
Administrative status for statistics collection.	**service stats**	Disabled
Multicast replication mode	**service multicast-mode**	head-end
VLAN translation	**service vlan-xlation**	Disabled
Administrative status of the service	**service admin-state**	Disabled

Refer to the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the above parameters and related commands.

Using VLAN Translation

VLAN translation refers to the egress translation of VLAN tags on service access points (SAPs). When enabled for a service, the VLAN tags for outgoing customer frames on SAPs associated with that service are processed according to the local SAP configuration (the SAP on which the frames will egress) and not according to the configuration of the SAP on which the frames were received.

- If the local SAP is configured for untagged traffic (*slot/port*:0), the egress traffic is always sent out as untagged.
- If the local SAP is configured for 802.1q-tagged traffic (*slot/port:ctag*), the egress traffic is single-tagged with the tag value specified by the *ctag* (customer VLAN tag) value.
- If the local SAP is configured for double-tagged traffic (*slot/port:outer_tag.: inner_tag*), the egress traffic is double-tagged with the tag values specified by the *outer_tag* and *inner_tag* values.

When VLAN translation is disabled, frames simply egress without any modification of the VLAN tags. In other words, the frames are transparently bridged without tag modification.

The following table shows the required translation (tag is added or replaced) that takes place when the egress SAP configuration is applied to the possible frame types (untagged, tagged, double-tagged). Note that in this table the terms “ITAG” and “OTAG” refer to inner tag and outer tag, respectively.

	Egress SAP (action required based on SAP type)		
	Untagged SAP	**Single Tagged SAP**	**Double-Tagged SAP**
Incoming Frame	Remove OTAG Remove ITAG	Replace OTAG **Note**: Replace = implicit add Remove ITAG	Replace OTAG **Note**: Replace = implicit add Add/Replace ITAG
Untagged	No tags, no action taken	Add the SAP OTAG	Add the SAP OTAG Add the SAP ITAG.
Single-tagged	Remove the OTAG	Replace the OTAG	Add ITAG Replace OTAG
Double-tagged	Remove the ITAG Remove the ITAG	Remove the ITAG Replace the OTAG	Replace ITAG Replace OTAG

Enabling VLAN translation is required at two different levels: first at the access port level and then at the service level. This activates VLAN translation for all of the SAPs on an access port that belong to the same service. The exception to this is on the OmniSwitch 9900, where enabling VLAN translation is required only at the service level.

Note. Mixing switches with VLAN translation enabled on some and disabled on other switches in the same network is not recommended. When VLAN translation is enabled on a service, all the frames forwarded by the service are treated as untagged within the service domain; the tag is inserted only during egress. As a result, VLAN translation should be enabled for the service across all of the participating switches to prevent frames from egressing the service domain as untagged frames.

To enable translation at the service level, use the **service vlan-xlation** command. For example:

```
-> service 1 vlan-xlation enable
```

To enable VLAN translation for all services, use the **all** parameter with the service spb command. For example:

```
-> service all vlan-xlation enable
```

To disable VLAN translation, use the **service vlan-xlation** command with the disable parameter. For example:

```
-> service 1 vlan-xlation disable
-> service all vlan-xlation disable
```

To enable VLAN translation at the access port level, use the **service access vlan-xlation** command. For example:

```
-> service access port 1/11 vlan-xlation enable
```

See "Configuring Service Access Ports" on page 7-56 for more information.

Enable the Service

By default, the SPB service is disabled when the service is created. Once the service is created and any optional service parameters are configured, use the **service admin-state** command with the **enable** option to enable the service. For example:

```
-> service 1 admin-state enable
```

To disable the service, enter the following command:

```
-> service 1 admin-state disable
```

Deleting an SPB Service

Before deleting a service from the switch configuration, disable the administrative status of the service. Once this is done, use the **no** form of the **service spb** command to delete the service. For example:

```
-> service 1 admin-state disable
-> no service 1 spb
```

Verifying the SPB Service Configuration

To view the SPB service configuration for the switch, use the **show service** command with the **spb** parameter option. For example:

```
-> show service spb
Legend: * denotes a dynamic object
SPB Service Info
  SystemId : 00e0.b1e7.0188,   SrcId : 0x70188,    SystemName : BEB-1

                              SAP     Bind                     MCast
ServiceId   Adm  Oper Stats Count   Count   Isid      BVlan Mode     (T/R)
-----------+----+----+-----+-------+-------+---------+-----+--------------
1           Up   Up    N    4       1       1000      4001  Headend  (0/0)
2           Up   Up    N    4       1       1001      4001  Headend  (0/0)
3           Up   Up    N    4       1       1002      4001  Headend  (0/0)
4           Up   Up    N    4       1       1003      4001  Headend  (0/0)
5           Up   Up    N    4       1       1004      4001  Headend  (0/0)
6           Up   Up    N    4       1       1005      4001  Headend  (0/0)
7           Up   Up    N    4       1       1006      4001  Headend  (0/0)
8           Up   Up    N    4       1       1007      4001  Headend  (0/0)
9           Up   Up    N    4       1       1008      4001  Headend  (0/0)
10          Up   Up    N    4       1       1009      4001  Headend  (0/0)
```

To view the configuration for an individual service, use the **show service spb** command and specify the SPB service ID. For example:

```
-> show service spb 1
SPB Service Detailed Info
  Service Id       : 1,                    Description      : ,
  ISID             : 1000,                 BVlan            : 4001,
  Multicast-Mode   : Headend,              Tx/Rx Bits       : 0/0,
  Admin Status     : Up,                   Oper Status      : Up,
  Stats Status     : No,                   Vlan Translation : No,
  Service Type     : SPB,                  Allocation Type  : Static,
  MTU              : 9194,                 Def Mesh VC Id   : 1,
  SAP Count        : 4,                    SDP Bind Count   : 1,
  Ingress Pkts     : 0,                    Ingress Bytes    : 0,
  Egress Pkts      : 0,                    Egress Bytes     : 0,
  Mgmt Change      : 08/10/2012 13:14:43,  Status Change    : 08/10/2012 13:14:00
```

Configuring Service Access Points (SAPs)

A SAP identifies the location where customer traffic enters the Provider Backbone Bridge Network (PBBN) edge, the type of customer traffic to service, parameters to apply to the traffic, and the service that will process the traffic for tunneling through the provider network.

Configuring a SAP requires several steps. These steps are outlined here and further described throughout this section:

- Configure customer-facing ports or link aggregates as service access ports.
- Configure Layer 2 profiles to determine how control packets are processed on access ports.
- Create a SAP by associating a SAP ID with an SPB service ID. A SAP ID is comprised of an access port and an encapsulation value, which is used to identify the type of customer traffic (untagged, single-tagged, or double-tagged) to map to the associated service.

SAP Configuration Guidelines

Consider the following when configuring a SAP:

- A SAP is a unique local entity for any given device. The same SAP ID value can be used on other BEB switches.
- There are no SAPs configured by default; explicit configuration of a SAP is required.
- A SAP is administratively disabled at the time the SAP is created.
- When a SAP is deleted, all configuration parameters for the SAP are also deleted.
- A SAP is owned by and associated with the service that was specified at the time the SAP was created.
- Multiple SAPs with different service types, such as a Virtual eXtensible LAN (VXLAN) or an SPB service, are allowed on the same service access port. For example, the following **show service access** command output shows two SAPs for port 2/1/30: one SAP bound to a VXLAN service and the other SAP bound to an SPB service:

```
-> show service access port 2/1/30 sap
Legend: * denotes a dynamic object
                                                                Vlan
Identifier    Adm  Oper Stats T:P  ServiceId   Isid/Vnid Xlation Sap Description
-------------+----+----+-----+----+-----------+---------+-------+---------------
sap:2/1/30A:0 Down Down  N    Y:x  20          1500        N        -
sap:2/1/30A:5 Up   Down  N    Y:x  10          23000       N        -

Total SAPs: 2
```

- If a port is administratively shutdown, all SAPs on that port become operationally out of service.
- Both fixed ports and link aggregates are configurable as access ports. Only access ports are associated with SAPs.
- Bridging functionality is not supported on access ports or link aggregates. In addition, if the default VLAN for a port or link aggregate is configured with an IP interface, then configuring the port or link aggregate as an access port is not supported.
- Configuring multiple SAPs on an access port that map different VLAN tags to the same service can cause a MAC move when the same customer MAC (CMAC) ingresses the access port with different VLAN tags. For example, a CMAC has two flows tagged with VLAN 10 and VLAN 20 ingressing access port 1/1 and both are mapped to service 100.

  ```
  -> service 100 sap port 1/1/1:10
  -> service 100 sap port 1/1/1:20
  ```

  To avoid the MAC move in this scenario, use one of the following alternative SAP configurations.

  Configure the SAPs with different services:

  ```
  -> service 100 sap port 1/1/1:10
  -> service 200 sap port 1/1/1:20
  ```

  Configure a default SAP to classify both flows into the same service:

  ```
  -> service 100 sap port 1/1/1:all
  ```

  See “Creating the Service Access Point” on page 7-60 for more information.

Configuring Service Access Ports

Each SAP is comprised of an access port or link aggregate and an encapsulation type value. Access ports are customer-facing ports that reside on a provider edge router. Traffic received on these ports is classified for one or more SAPs and forwarded onto the intended destination by the associated SPB service.

To configure a port or link aggregate as an access port, use the **service access** command. For example, the following command configures port 1/1/2 and link aggregate 5 as access ports:

```
-> service access port 1/1/2
-> service access linkagg 5
```

To revert an access port back to a regular switch port or link aggregate, use the **no** form of the **service access** command. For example:

```
-> no service access port 1/1/2
-> no service access linkagg 5
```

VLAN Translation on Access Ports

VLAN translation refers to the egress translation of VLAN tags on service access points (SAPs). For more information about how VLAN translation is applied, see "Using VLAN Translation" on page 7-53.

By default, VLAN translation is disabled on access ports. Enabling VLAN translation on an access port implicitly enables translation for all SAPs associated with that port. However, translation must also be enabled for the services associated with these SAPs. This ensures that all SAPs associated with a service will apply VLAN translation.

Note. On the OmniSwitch 9900, enabling VLAN translation at the access port level is not required. See "Using VLAN Translation" on page 7-53 for more information.

To enable VLAN translation on an access port, use the **service access vlan-xlation** command with the **enable** option. For example:

```
-> service access port 1/1/3 vlan-xlation enable
-> service access linkagg 10 vlan-xlation enable
```

To disable VLAN translation on an access port, use the **service access vlan-xlation** command with the **disable** option. For example:

```
-> service access port 1/1/3 vlan-xlation disable
-> service access linkagg 10 vlan-xlation disable
```

Configuring Layer 2 Profiles for Access Ports

A Layer 2 profile determines how control frames ingressing on an access port are processed. When a port is configured as an access port, a default Layer 2 profile (**def-access-profile**) is applied to the port with the following default values for processing control frames:

Protocol	Default
STP	tunnel
802.1x	drop
802.3ad	peer
MVRP	tunnel
GVRP	tunnel
AMAP	drop
802.1ab	drop

If the default profile values are not sufficient, use the **service l2profile** command with the **tunnel**, **drop**, and **peer** options to create a new profile. For example, the following command creates a profile named "DropL2":

```
-> service l2profile DropL2 stp drop gvrp drop 802.1ab drop
```

Consider the following when configuring Layer 2 profiles:

- When a profile is created, the new profile inherits the default profile settings for processing control frames. The default settings are applied with the new profile unless they are explicitly changed. For example, the profile "DropL2" was configured to discard STP, GVRP, and 802.1ab frames. No other protocol settings were changed, so the default settings still apply for the other protocols.

- Remove any profile associations with access ports before attempting to modify or delete the profile.
- Not all of the control protocols are currently supported with the **peer**, **tunnel**, and **drop** parameters. Use the following table to determine the parameter combinations that are supported:

Protocol	Reserved MAC	peer	drop	tunnel
STP	01-80-C2-00-00-00	no	yes	yes
802.1x	01-80-C2-00-00-03	yes	yes	yes
802.1ab	01-80-C2-00-00-0E	yes	yes	yes
802.3ad	01-80-C2-00-00-02	yes	no	no
GVRP	01-80-C2-00-00-21	no	yes	yes
AMAP	00-20-DA-00-70-04	yes	yes	yes
MVRP	01-80-C2-00-00-21	no	yes	yes

To delete a user-created Layer 2 profile, use the **no** form of the **service l2profile** command. For example, the following command deletes the "DropL2" profile:

```
-> no service l2profile DropL2
```

Use the **show service l2profile** command to view a list of profiles that are already configured for the switch. This command also displays the attribute values for each profile.

Configuring a Layer 2 Profile Action for 802.1AB PDUs

A Layer 2 profile can be configured to apply a different action to tagged and untagged 802.1AB control frames. For example, the following command uses the **service l2profile inbound 802.1ab** command with the **tagged** parameter to set the action for tagged 802.1AB control frames in the specified Layer 2 profile:

```
-> service l2profile lldp-tagged inbound tagged 802.1ab tunnel
```

In this example, the tunnel action is specified only for tagged 802.1AB control frames. All tagged 802.1AB control frames will be tunneled, while untagged frames will be dropped.

To configure an action for untagged 802.1AB control frames, use the **service l2profile inbound 802.1ab** command with the **untagged** parameter to set the action for untagged 802.1AB control frames in the specified Layer 2 profile. For example:

```
-> service l2profile lldp-untagged inbound untagged 802.1ab peer
```

In this example, the peer action is specified only for untagged 802.1AB control frames. All untagged 802.1AB control frames will participate in the protocol, while tagged frames will be dropped.

To configure the same action for both tagged and untagged 802.1AB control frames, use the **service l2profile inbound 802.1ab** command with the **both** parameter. For example:

```
-> service l2profile lldp-both inbound both 802.1ab peer
```

When a UNI profile is configured to apply a different action for tagged and untagged 802.1AB PDUs, the profile action can only be modified through one of the following methods:

- Set both the tagged and untagged action for 802.1AB PDUs back to the default setting (**drop**) then configure a new action for both.

- Delete the UNI profile and create a new one with the modified action for tagged and untagged 802.1AB PDUs.

Assigning Layer 2 Profiles to Access Ports

After a Layer 2 profile is created, it is then necessary to assign the profile to an access port or link aggregate. When this is done, the current profile associated with the port is replaced with the new profile.

The **service access l2profile** command is used to assign a new profile to an access port. For example, the following command assigns the "DropL2" profile to access port 1/1/4 and link aggregate 5:

```
-> service access port 1/1/4 l2profile DropL2
-> service access linkagg 5 l2profile DropL2
```

To change the profile associated with the access port back to the default profile (**def-access-profile**), use the **default** option with the **service access l2profile** command. For example:

```
-> service port 1/1/4 l2profile default
-> service access linkagg 5 l2profile default
```

When a service access port is configured with the **def-access-profile** and then assigned to an SPB pseudo-wire SAP, the Layer 2 profile for that port automatically changes to **pw-def-access-profile** with the following control protocol settings (see "Configuring an SPB Pseudo-Wire Service" on page 7-63 for more information):

parameter	def-access-profile
stp	**tunnel**
802.1x	**tunnel**
802.3ad	**peer**
mvrp	**tunnel**
gvrp	**tunnel**
amap	**tunnel**
802.1ab	**tunnel**

- If a user-created Layer 2 profile is associated with a service access port that is then assigned to an SPB pseudo-wire SAP, the Layer 2 profile for that port is *not* changed. The user-created profile is applied to control packets received on the port, not the **pw-def-access-profile**.
- When the **default** option is used with the **service access l2profile** command, the Layer 2 profile assignment for the service access port reverts back to **def-access-profile**, regardless of whether the port was assigned to a user-created profile or to **pw-def-access-profile**.

Verifying the Access Port Configuration

To view the access port configuration for the switch, use the **show service access** command. For example:

```
-> show service access
Port        Link  SAP     SAP     Vlan
Id         Status Type    Count   Xlation L2Profile
---------+------+-------+-------+-------+-------------------
1/1/3      Up     Manual  100     N       def-access-profile
1/1/4      Down   Manual  100     N       def-access-profile
1/1/5      Down   Manual  100     N       def-access-profile
1/1/15     Down   Dynamic 0       Y       def-access-profile
```

```
1/1/16    Up     Dynamic 1       Y       def-access-profile
1/1/17    Down   Dynamic 0       Y       def-access-profile

Total Access Ports: 6
```

Creating the Service Access Point

Each SPB service is bound to at least one Service Access Point (SAP). A SAP identifies the point at which customer traffic enters the Provider Backbone Bridge Network (PBBN). Creating a SAP on an SPB switch designates that switch as a Backbone Edge Bridge (BEB) in the PBBN. An SPB switch that does not have a SAP but does have at least one BVLAN and an SPB interface is designated as Backbone Core Bridge (BCB) in the PBBN.

Once the SPB topology is determined and switches that will serve as BEBs are identified, a SAP is created on each BEB. A SAP is created by associating a SAP ID with an SPB service. A SAP ID is comprised of a customer-facing port (referred to as an access port) and an encapsulation value that is used to identify the type of customer traffic (untagged, single-tagged, or double-tagged) to map to the associated service.

The **service sap** command is used to configure a SAP. This command specifies the SPB service ID number and the SAP ID (slot/port:encapsulation). The following parameter values are used with this command to specify the encapsulation value:

SAP Encapsulation Value	Customer Traffic Serviced
0 (null)	All untagged packets; tagged packets are dropped.
all	All tagged and untagged packets not already classified into another SAP*
qtag	Only traffic 802.1q-tagged with the specified VLAN ID.
outer_*qtag.inner_qtag*	Only traffic double-tagged (QinQ) with the specified outer and inner VLAN IDs.

*Note that the **:all** (wildcard) parameter is also configurable as the inner tag value for double-tagged frames (for example, “10:all” specifies double-tagged packets with an outer tag equal to 10 and an inner tag with any value).

The following **service sap** command example creates a SAP that will direct customer traffic ingressing on access port 1/1/4 that is tagged with VLAN ID 50 to service 100:

```
-> service 100 sap 1/1/4:50 description “BEB1 to SPB100 CVLAN 50”
```

In the above example, the 1/1/4:50 designation is referred to as the SAP ID or the encapsulation ID. This means that if no other SAPs are configured for port 1/1/4, then any traffic ingressing on that port is dropped if the traffic is not tagged with VLAN 50.

It is possible to configure more than one SAP for the same access port, which provides a method for segregating incoming traffic into multiple services. For example, the following SAP configuration for port 2/1/3 sends incoming traffic to three different services based on the VLAN tags of the frames received:

```
-> service 2000 sap port 2/1/3:all
-> service 200 sap port 2/1/3:100
-> service 1000 sap port 2/1/3:100.200
```

In this example,

- Frames double-tagged with 100 (outer tag) and 200 (inner tag) are sent on service 1000.

- Frames single-tagged with VLAN 100 are sent on service 200.
- All other frames (those that are not single-tagged with 100 or double-tagged with 100 and 200) are sent on service 2000.

The following SAP ID classification precedence is applied when there are multiple SAPs for one access port:

1. Double-tagged (Outer VLAN + Inner VLAN) - Highest
2. Double-tagged (Outer VLAN + all)
3. Single-tagged (VLAN)
4. Single-tagged (wildcard)
5. Untagged - Lowest.

Modifying Default SAP Parameters

The following parameter values are set by default at the time the SAP is created. If necessary, use the specified commands to change the default values.

Parameter Description	Command	Default
SAP description.	**service sap description**	None
SAP trust mode	**service sap trusted**	Trusted
Administrative status for the SAP	**service sap admin-state**	Enabled
Administrative status for statistics collection.	**service sap stats**	Disabled

Refer to the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the command parameters.

Configuring the SAP Trust Mode

The **service sap trusted** command is used to configure the trust mode for a SAP. A trusted SAP can accept 802.1p values in incoming packets; an untrusted SAP will set any 802.1p values to zero in incoming packets, unless an 802.1p value is configured with this command.

Note that untagged Layer 2 control packets (for example, BPDU, GVRP, and AMAP) are always tunneled (if enabled) through the Provider Backbone Bridge (PBB) network with the default EXP bits set to 7, so that they can arrive at the destination bridge at the highest COS queue of 7. As a result, trusted and untrusted SAPs configured on the access ports will not affect the Layer 2 control packets ingressing on the access ports.

By default, a SAP is trusted with the priority set to best effort (zero). Use the **no** form of the **service spb sap trusted** command with the **priority** option to change the SAP mode to untrusted. For example:

```
-> service 100 sap 1/4:50 no trusted priority 7
```

When a SAP is trusted, the priority value contained in tagged customer packets is used; untagged packets are assigned the default priority value (zero). When a SAP is untrusted, the priority value configured for the SAP is assigned to both tagged and untagged customer packets.

Enabling/Disabling the SAP

By default, a SAP is disabled at the time the SAP is created. To enable the SAP administrative status, use the **service sap admin-state** command. For example:

```
-> service 100 sap port 1/4:50 admin-state enable
-> service 200 sap linkagg 5:all admin-state enable
```

To disable the SAP, enter the following command:

```
-> service 100 sap port 1/4:50 admin-state disable
-> service 200 sap linkagg 5:all admin-state disable
```

Deleting the SAP

When a SAP is administratively disabled, the SAP configuration is not removed from the switch. To delete a SAP from the switch configuration, use the **no** form of the **service sap** command. For example:

```
-> service 100 no sap port 1/4:50
-> service 200 no sap linkagg 5:all
```

Verifying the SAP Configuration

A SAP is a type of virtual port that is associated with an SPB service. To determine the SAP configuration for a specific service, use the **show service ports** command to view the virtual ports associated with a specific service. For example:

```
-> show service 1 ports
Legend: * denotes a dynamic object
SPB Service Info
  Admin : Up, Oper  : Up, Stats      : N, Mtu     : 9194,   VlanXlation : N,
  ISID  : 1000,      BVlan : 4001,   MCast-Mode : Headend,   Tx/Rx   : 0/0

                                Sap Trusted:Priority/         Sap Description /
Identifier       Adm  Oper Stats Sdp SystemId:BVlan   Intf     Sdp SystemName
----------------+----+----+-----+--------------------+--------+--------------------
sap:1/11:1000    Up   Up    N           Y:x           1/11        -
sap:1/12:1000    Up   Down  N           Y:x           1/12        -
sap:1/13:1000    Up   Down  N           Y:x           1/13        -
sap:1/14:1000    Up   Down  N           Y:x           1/14        -
sdp:32776:1*     Up   Up    Y    e8e7.3233.1831:4001  1/1        BEB-1

Total Ports: 5
```

To then view configuration information for a specific SAP, use the **show service spb sap** command. For example:

```
-> show service 1 sap port 1/11:1000
SAP Detailed Info
  SAP Id           : 1/11:1000,            Description      : ,
  Admin Status     : Up,                   Oper Status      : Up,
  Stats Status     : No,                   Vlan Translation : No,
  Service Type     : SPB,                  Allocation Type  : Static,
  Trusted          : Yes,                  Priority         : 0,
  Ingress Pkts     : 0,                    Ingress Bytes    : 0,
  Egress Pkts      : 0,                    Egress Bytes     : 0,
  Mgmt Change      : 08/07/2012 23:39:29,  Status Change    : 08/10/2012 15:13:08
```

Configuring an SPB Pseudo-Wire Service

The OmniSwitch implementation of SPB provides a multipoint-to-multipoint (E-LAN) connectivity solution that connects customer sites across an SPB network. However, as the number of users increases, the number of MAC addresses learned increases and more system resources are used up. It is also possible that a simple point-to-point virtual wire (E-LINE) connection between two SPB nodes is all that is required.

An SPB pseudo-wire service provides a point-to-point connection (E-LINE) that will bridge Ethernet traffic between two specifically designated end points. Configuring an SPB pseudo-wire service is done as part of defining an end point for a point-to-point connection between two SAPs (also referred to as attachment points for a pseudo-wire).

With a pseudo-wire point-to-point connection, there is no forwarding decision to be made; packets simply enter one end of the connection and leave the other end of the connection unchanged. As a result, customer MAC addresses are not learned on the SAP attachment points. Packets are transparently forwarded, which simplifies traffic flow and reduces hardware usage.

When configuring an SPB pseudo-wire service, as described in "SPBM Pseudo-Wire (E-LINE Transparent) Service" on page 7-14, consider the following guidelines:

- The SAP that the service is associated with serves as an attachment point for one end of the point-to-point connection.
  - If the other end of the connection is to a local SAP, make sure both SAPs are associated with the same service.
  - If the other end of the connection is to a remote SAP, make sure both the local and remote SAP are associated with the same I-SID. This will ensure that there is only one virtual wire connection between the two SPB nodes.
  - Make sure there are no more than two SAP attachment points for the same pseudo-wire service.
- Do not configure a pseudo-wire service with the same I-SID on more than two SPB nodes. Configuring the same pseudo-wire service on more than two nodes may cause unpredictable behavior with the connection.
- Configure the SPB pseudo-wire service to use the head-end multicast mode.
- VLAN translation is automatically disabled to ensure packets enter and leave the pseudo-wire connection with the original customer VLAN tag.
- Use the **all** option for the SAP encapsulation (1/1/2:all) to ensure all customer traffic flows through the pseudo-wire connection.
- Flooding and replication are not needed, as there are only two attachment points (SAP to SAP for a local service or SAP to an SPB network port for the same service instance across the SPB backbone network).
- IPv4 over SPB routing, IPv6 over SPB routing, and IGMP snooping are not supported on an SPB pseudo-wire service.
- When an access port that is associated with the Layer 2 profile **def-access-profile** is assigned to an SPB pseudo-wire SAP attachment point, the Layer 2 profile for that port is automatically changed to **pw-def-access-profile**. See the "Assigning Layer 2 Profiles to Access Ports" on page 7-59 for more information.

To configure an SPB pseudo-wire connection, identify which SPB nodes will serve as a pseudo-wire endpoint. This usually is a node that connects to a customer edge switch. Using the **service spb** command with the **pseudo-wire** option, configure an SPB service as a pseudo-wire type service. For example:

```
-> service 100 spb isid 1000 bvlan 4001 pseudo-wire enable description "Pseudo-
wire ISID 1000"
```

After the service is created, use the **service sap** command to configure a SAP for the service and specify the **all** option for the SAP encapsulation. For example:

```
-> service 100 sap port 1/1/10:all
```

Verifying the SPB Pseudo-Wire Configuration

To verify the type of SPB service that is configured, use the **show service** command and specify the service ID. For example, the following commands displays information about SPB service 100:

```
-> show service 100
SPB Service Detailed Info
  Service Id       : 100,                 Description      : Pseudo-wire ISID 1000,
  ISID             : 1000,                BVlan            : 4001,
  Multicast-Mode   : Headend,             Tx/Rx Bits       : 0/0,
  Admin Status     : Up,                  Oper Status      : Up,
  Stats Status     : No,                  Vlan Translation : No,
  Service Type     : SPB,                 Allocation Type  : Static,
  MTU              : 9194,                VPN IP-MTU       : 1500,
  SAP Count        : 1,                   SDP Bind Count   : 1,
  RemoveIngressTag : No,                  Option           : Pseudo-wire,
  Mgmt Change      : 12/05/2015 18:42:43, Status Change    : 12/05/2015 18:42:43
```

The “Option” field indicates that service 100 is a pseudo-wire service (tunnel endpoints are manually configured to define a single point-to-point tunnel connection between two SPB attachment points). If this field displays **None**, the service is operating in the default multipoint-to-multipoint mode (tunnel endpoints are automatically defined between all the SPB nodes with the same I-SID.

To verify the virtual ports associated with a pseudo-wire service, use the **show service ports** command. For example, the following command displays the virtual port information for pseudo-wire service 100:

```
-> show service 100 ports
Legend: (*)Dyn unicast (+)Remote Mcast (#)Local Mcast (-)ERP Ring
SPB Service 100 Info
 Admin : Up,   Oper  : Down, Stats      : Y,       Mtu     : 9194, VlanXlation : N,
 ISID  : 1000, BVlan : 4001, MCast-Mode : Headend, Tx/Rx   : 0/0,  RemoveIngTag: N,
 PW-Enb: Y,

                                 Sap Trusted:Priority/         Sap Description /
Identifier        Adm  Oper Stats Sdp SystemId:BVlan   Intf     Sdp SystemName
-----------------+----+----+-----+--------------------+--------+-------------------
sap:1/1/5:all     Up   Up    N           Y:x           1/1/5       -
sdp:32768:100*    Up   Up    N    e8e7.32f8.5b41:4000  0/120    X72_VC

Total Ports: 2
```

MAC learning is automatically disabled for SAP 1/1/5:all and SDP 32768:100, because service 100 defines an endpoint for a pseudo-wire connection.

SPB Pseudo-Wire Configuration Example

The following diagram shows an example of a pseudo-wire point-to-point connection between two remote sites across the SPB network cloud. In this sample topology,

- Customer Edge-1 (CE-1) carries different traffic flows from its site to the attachment point SAP on BEB-1, which is configured as an endpoint for the pseudo-wire. The traffic is then transparently forwarded across the SPB network towards the other end of the pseudo-wire on BEB-2.

- Customer-Edge-2 receives the traffic flows from CE-1 in the original format.
- As the number of users increases between the two customer sites, the number of MAC addresses does not increase, as there is no MAC address learning on the pseudo-wire SAP attachment points or the SPB network ports on BEB-1 and BEB-2.

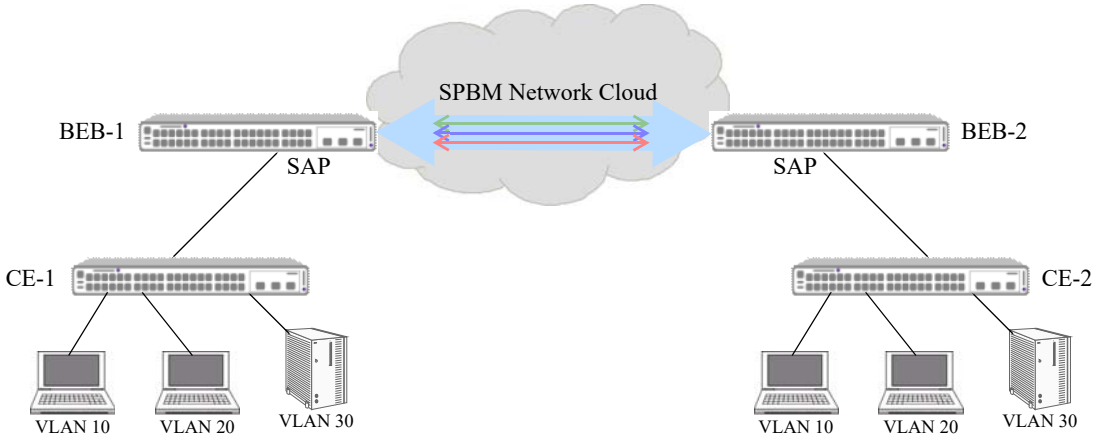


Figure 7-14 : Remote SAP Pseudo-Wire Connection

The two sites (CE-1 and CE-2) are connected through the pseudo-wire as if they were on the same single line (E-LINE). The following diagram shows a logical depiction of how traffic flows between the devices that are remotely connected through the pseudo-wire as if they were locally connected.

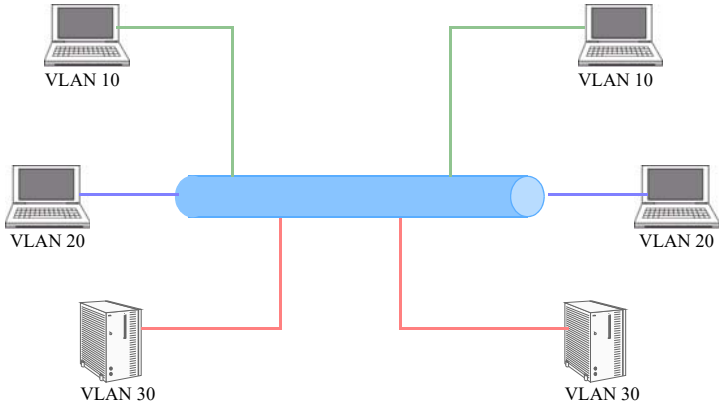


Figure 7-15 : Devices Remotely Connected on a Pseudo-Wire

The following CLI command examples are used to configure BEB-1 and BEB-2 in the sample SPB pseudo-wire topology:

```
-> spb bvlan 4001
-> spb isis control-bvlan 4001
-> spb isis interface port 1/1/5
-> service access port 1/1/10
-> service 100 spb isid 1000 bvlan 4001 pseudo-wire enable description "Pseudo-
wire for ISID 1000"
-> service 100 sap port 1/1/10:all
-> spb isis admin-state enable
```

Configuring Remote Fault Propagation for SPBM

Remote Fault Propagation (RFP) for SPBM monitors SPB access ports to detect link failures that cause interruptions to SPB services. The status of an access port and any associated I-SID is advertised within an RFP domain using Continuity Check Message (CCM) packets. When a CCM packet is received that indicates an access port for a specific I-SID is down, the corresponding access port associated with the same I-SID is automatically taken down.

Consider the following recommended guidelines when configuring RFP for SPBM:

- Configuring an RFP domain involves creating a local Maintenance End Point (MEP) on each switch that will participate in the RFP domain. The MEP is mapped to a reserved Ethernet OAM domain. This type of domain counts towards the maximum limit of Ethernet OAM domains allowed.
- The SPB Control BVLAN serves as the primary VLAN for all RFP domains. CCM packets are sent across the SPBM network to all BEB devices on the Control BVLAN. However, CCM packets are not encapsulated with SPB header information.
- Make sure to use the same CCM interval value for all local MEPs that participate in the same RFP domain. A mismatch will prevent reliable communication between MEPs.
- The SPB service associated with the I-SID that RFP will monitor should be configured on only two Backbone Edge Bridges (BEBs) in the network.
- The SPB service associated with an I-SID is mapped to only one SAP. For example, SPB service 10 bound to I-SID 1500 is mapped only to a SAP configured on port 1/12; service 10 is not mapped to any other SAP on the same switch.
- Configure only one SAP on a physical access port; configuring additional SAPs on the same port is not recommended.
- Configure a SAP associated with an RFP monitored port on only one physical port of the BEB.

Configuring an RFP Domain

A local Maintenance End Point (MEP) and a corresponding remote end point list are configured on each BEB that will participate in an RFP domain. The domain to which each end point is assigned is determined by the RFP domain ID associated with each end point. The domain ID is defined at the time the local MEP is created using the **service rfp local-endpoint** command. For example:

```
-> service rfp 1 local-endpoint 10 type spb
```

In this example, RFP domain 1 is created with a local MEP ID of 10. By default, the CCM interval is set to 1 second, the domain level is set to 7, and the administrative status is enabled for RFP domain 1. To set different values for these parameters, use the **service rfp local-endpoint** command with the **ccm-interval**, **level**, or **admin-state** parameters. For example:

```
-> service rfp 1 local-endpoint 10 ccm-interval interval10s level 6 admin-state
disable type spb
```

In this example, RFP domain 1 with local MEP ID 10 is created with the CCM interval set to 10 seconds, the domain level set to 6, and the administrative status disabled.

The parameter values used to created an RFP domain are used to create a reserved Ethernet OAM domain on the local switch to which the RFP domain is mapped. The reserved OAM domain is given the name “RFP_OVER_SPB_DOMAIN_LEVELx”, where x is the number specified with the **level** parameter. It is important to note that each RFP domain created must use a different level number. For example, if RFP domain 1 uses level 7, then RFP domain 2 must use a different level number (for example, level 6).

All the reserved OAM domains that are automatically created for RFP domains are assigned to the same "RFP_OVER_SPB_ASSOCIATION" Maintenance Association (MA).

Use the **show service rfp configuration** to display the Ethernet OAM domain parameters configured and associated with the RFP domain. For example, the following shows the OAM domain reserved for the RFP 1 domain ID:

```
-> show service rfp configuration
Total Number of RFP domains - 1

RFP Domain Number       : 1
Admin Status            : Enabled
Level                   : 7
Type                    : SPB
Maintenance Domain      : RFP_OVER_SPB_DOMAIN_LEVEL7
Maintenance Association : RFP_OVER_SPB_ASSOCIATION
Control B-VLAN          : 500
Virtual UP MEP ID       : 10
CCM Interval            : 10 minutes
Remote Endpoint           Service Id
------------------------+-------------
```

In this example, the "Remote Endpoint" and "Service Id" fields are blank because a list of remote end points and SPB services has not yet been created for RFP domain 1. A remote end point list provides a list of remote MEP IDs (local MEP IDs configured on other BEBs) and a list of SPB service IDs that are active on the local BEB. CCM packets are sent to all the remote MEP IDs on the list to advertise the status of the local SAP ports and I-SIDs associated with the specified SPB services.

To create a remote end point list for RFP domain 1, use the **service rfp remote-endpoint** command. For example:

```
-> service rfp 1 remote-endpoint 2 service-id 10-12
```

In this example, remote end point 2 is the MEP ID that identifies a remote BEB that is participating in the same RFP domain. The service IDs 10, 11, and 12 are the SPB services bound to the RFP 1 domain.

Creating a remote end point list triggers the transmission of CCM packets carrying I-SID and port status information related to the specified SPB services. Configure a different remote end point for each remote BEB that needs to receive the CCM packet information for a specific service. For example, the following commands add MEP ID 3 and 4 as remote end points to receive status for services 13 and 14:

```
-> service rfp 1 remote-endpoint 3 service-id 13
-> service rfp 1 remote-endpoint 4 service-id 14
```

Use the **show service rfp configuration** to display the RFP Ethernet OAM domain configuration showing the remote end points added to RFP domain 1. For example:

```
-> show service rfp configuration
Total Number of RFP domains - 1

RFP Domain Number       : 1
Admin Status            : Enabled
Level                   : 7
Type                    : SPB
Maintenance Domain      : RFP_OVER_SPB_DOMAIN_LEVEL7
Maintenance Association : RFP_OVER_SPB_ASSOCIATION
Control B-VLAN          : 500
Virtual UP MEP ID       : 10
CCM Interval            : 10 minutes
```

```
Remote Endpoint           Service Id
------------------------+-------------
2                          10
2                          11
2                          12
3                          13
4                          14
```

Deleting the RFP Domain

To remove an RFP domain configuration from the switch, use the following command**:**

```
-> no service rfp 1
```

The above command removes all of the RFP configuration items, including the reserved Ethernet OAM domain for the specified RFP ID.

Modifying the local MEP ID

To change a local end point (MEP ID), first set the ID to zero using the **no** form of the **service rfp local-endpoint** command and specify the existing ID number. For example:

```
-> no service rfp 1 local-endpoint 10
```

Next, set the local MEP ID to a different value. For example, the following command sets a new value of 15 for the local MEP ID:

```
-> service rfp 1 local-endpoint 15
```

Removing an SPB Service from RFP Monitoring

To discontinue RFP monitoring of SPB service instances within an RFP domain, remove the local MEP ID from the remote end point list on each BEB that terminates the service. To remove a MEP ID from a remote end point list, use the **no** form of the **service rfp remote-endpoint** command. For example:

```
-> no service rfp 1 remote-endpoint 2
```

In this example, MEP ID 2 is removed from the remote end point list along with all SBP services associated with MEP ID 1. RFP will no longer monitor and advertise the status of local services to remote MEP ID 2.

To remove a specific SPB service from a remote end point list, use the **no** form of the **service rfp remote-endpoint** command with the **service-id** parameter. For example, the following command removes SPB service 10 associated with MEP ID 2:

```
-> no service rfp 1 remote-endpoint 2 service-id 10
```

In this example, SPB service 10 was removed from the end point list for MEP ID 2. However, RFP will continue to monitor and advertise all other services to this remote end point.

It is important to consider that when a MEP ID or a specific SPB service ID is removed from the end point list on the local switch, a port violation will occur. This can cause an undesirable service interruption when attempting to simply unbind a service from an RFP domain. To avoid a port violation condition, remove the SPB service ID from both ends of the SPB tunnel at the same time.

Verifying the RFP for SPB Configuration

To verify the connectivity between remote end points within an RFP domain, use the **show service rfp** command. For example:

```
-> show service rfp

Local system (Name : SystemId) =  Edge-39 : e8e7.326c.4a39
Total number of services information  = 2
Total number of RFP domain  = 3
RFP    Remote   RMEP      System                      B-VLAN ISID    Service Admin
       EndPoint Status    (Name : SystemId)                          Id      State
------+--------+--------+-------------------------+------+-------+-------+--------
1      2        RMEP_OK  Edge-43: 00:e0:b1:e7:09:a3 500    1001    10      Enabled
1      2        RMEP_OK  Edge-43: 00:e0:b1:e7:09:a3 500    1002    11      Enabled
1      2        RMEP_OK  Edge-43: 00:e0:b1:e7:09:a3 500    1003    12      Enabled
```

To verify the status of the local SAP associated with the RFP domain, use the **show service rfp** command with the **local-sap-status** parameter. For example:

```
-> show service rfp 1 local-sap-status

Local endpoint ID = 10
Local system (Name : SystemId)  = Edge-39 : e8e7.326c.4a39
Service  Sap                Admin         Oper           Remote         R-Endpoint
Id                                                       Endpoint       Status
--------+------------------+-------------+--------------+--------------+-----------
10       sap:1/12:all       Enabled       Down           2              RMEP_OK
11       sap:1/11:all       Enabled       Down           2              RMEP_OK
12       sap:1/10:all       Enabled       Down           2              RMEP_OK
```

As previously described, use the **show service rfp configuration** command to display the parameter values associated with the RFP domain. For example:

```
-> show service rfp configuration
Total Number of RFP domains - 1

RFP Domain Number       : 1
Admin Status            : Enabled
Level                   : 7
Type                    : SPB
Maintenance Domain      : RFP_OVER_SPB_DOMAIN_LEVEL7
Maintenance Association : RFP_OVER_SPB_ASSOCIATION
Control B-VLAN          : 500
Virtual UP MEP ID       : 10
CCM Interval            : 10 minutes
Remote Endpoint           Service Id
------------------------+-------------
2                          10
2                          11
2                          12
```

RFP for SPB Configuration Example

This section contains CLI command examples used to configure the RFP domain functionality deployed in the following sample SPB network topology:

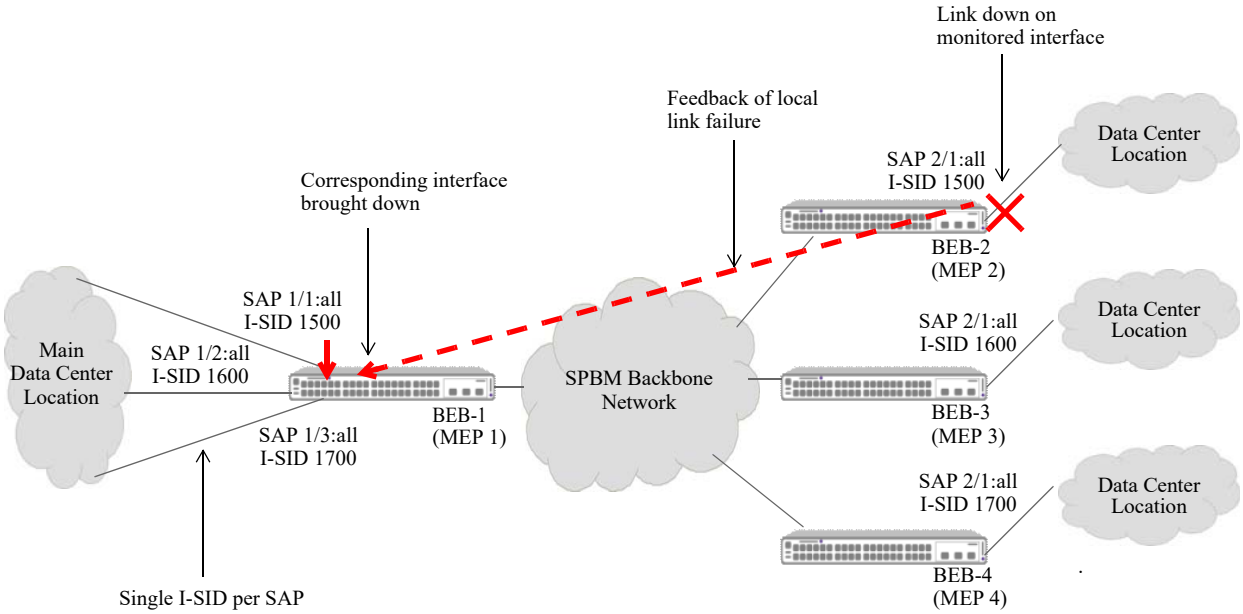


Figure 7-16 : RFP for SPB Example

In this topology, RFP domain 1 is created by configuring a Maintenance End Point (MEP) on each OmniSwitch serving as an SPB BEB. The local MEP ID associated with each switch (MEP 1, 2, 3, and 4) identifies that switch as a participant in the RFP domain. As shown in the above diagram:

- Three I-SIDs (1500, 1600, and 1700) are each bound to a separate SAP port (1/1, 1/2, and 1/3) on BEB-1. This represents one end of each SPB service.
- The other end of each SPB service is bound to SAP ports on BEB-2 (I-SID 1500), BEB-3 (I-SID 1600), and BEB-3 (I-SID 1700).
- The RFP configuration on BEB-1 specifies the MEP ID of the other three BEBs as remote end points to which the local status of the three I-SIDS and SAP ports is advertised using CCM packets.
- The RFP configuration on BEB-2 specifies the MEP ID of BEB-1 as a remote end point to which the local status of I-SID 1500 and SAP port 2/1 is advertised using CCM packets.
- The RFP configuration on BEB-3 specifies the MEP ID of BEB-1 as a remote end point to which the local status of I-SID 1600 and SAP port 2/1 is advertised using CCM packets.
- The RFP configuration on BEB-4 specifies the MEP ID of BEB-1 as a remote end point to which the local status of I-SID 1700 and SAP port 2/1 is advertised using CCM packets.
- When SAP port 2/1 goes down on BEB-2, the port down status is reported in the CCM transmitted from BEB-2 to BEB-1.
- When BEB-1 receives the CCM packet from BEB-2 and detects the port down status, BEB-1 administratively downs the corresponding SAP port 1/1. The service associated with I-SID 1500 stops on both ends of the service (BEB-1 and BEB-2).
- When the downed port on BEB-2 is brought back up, BEB-1 receives the port up status from BEB-2 and brings the local SAP port 1/1 back up as well.

The following CLI command examples include the SPB commands used to create the SPB service layer that RFP will monitor.

BEB-1:

```
-> spb bvlan 4001 admin-state enable
-> spb isis bvlan 4001 ect-id 2
-> spb isis control-bvlan 4001
-> spb isis interface port 1/20
-> spb isis admin-state enable
-> service access port 1/1
-> service access port 1/2
-> service access port 1/3
-> service spb 1 isid 1500 bvlan 4001 admin-state enable
-> service spb 2 isid 1600 bvlan 4001 admin-state enable
-> service spb 3 isid 1700 bvlan 4001 admin-state enable
-> service spb 1 sap port 1/1:all admin-state enable
-> service spb 2 sap port 1/2:all admin-state enable
-> service spb 3 sap port 1/3:all admin-state enable

-> service rfp 1 local-endpoint 1 ccm-interval interval100ms type spb
-> service rfp 1 remote-endpoint 2 service-id 1 admin-status enable
-> service rfp 1 remote-endpoint 3 service-id 2 admin-status enable
-> service rfp 1 remote-endpoint 4 service-id 3 admin-status enable
```

BEB-2:

```
-> spb bvlan 4001 admin-state enable
-> spb isis bvlan 4001 ect-id 2
-> spb isis control-bvlan 4001
-> spb isis interface port 1/21
-> spb isis admin-state enable
-> service access port 2/1
-> service spb 1 isid 1500 bvlan 4001 admin-state enable
-> service spb 1 sap port 2/1:all admin-state enable

-> service rfp 1 local-endpoint 2 ccm-interval interval100ms type spb
-> service rfp 1 remote-endpoint 1 service-id 1 admin-status enable
```

BEB-3:

```
-> spb bvlan 4001 admin-state enable
-> spb isis bvlan 4001 ect-id 2
-> spb isis control-bvlan 4001
-> spb isis interface port 1/22
-> spb isis admin-state enable
-> service access port 2/1
-> service spb 2 isid 1600 bvlan 4001 admin-state enable
-> service spb 2 sap port 2/1:all admin-state enable

-> service rfp 1 local-endpoint 3 ccm-interval interval100ms type spb
-> service rfp 1 remote-endpoint 1 service-id 2 admin-status enable
```

BEB-4:

```
-> spb bvlan 4001 admin-state enable
-> spb isis bvlan 4001 ect-id 2
-> spb isis control-bvlan 4001
-> spb isis interface port 1/23
-> spb isis admin-state enable
-> service access port 2/1
-> service spb 3 isid 1700 bvlan 4001 admin-state enable
-> service spb 3 sap port 2/1:all admin-state enable

-> service rfp 1 local-endpoint 4 ccm-interval interval100ms type spb
-> service rfp 1 remote-endpoint 1 service-id 3 admin-status enable
```

Configuring IP over SPB

Configuring IP over SPB, as described in "IP over SPBM" on page 7-18, requires the following general steps:

- Define an L3 VPN interface to serve as a gateway address to remote networks. The following options are available for configuring an L3 VPN interface (refer to the table on page 7-19 to determine which L3 VPN interface options are supported based on the switch platform):
  - An external loopback port configuration.
  - An IP service-based interface configured through software for single-pass inline routing.
  - Front-panel ports configured through software for two-pass inline routing.
- Determine whether to use the VPN-Lite or L3 VPN (ISIS-SPB) approach for routing L3 traffic over an L2 SPBM backbone network.
- To implement the VPN-Lite approach, configure dynamic routing protocols (such as OSPF) or static routes directly on the L3 VPN interface.
- To implement the L3 VPN (ISIS-SPB) approach, configure a binding between a VRF instance, an SPB I-SID, and an IP gateway (the L3 VPN interface address). This binding will enable bidirectional exchange of routes between the VRF and SPB I-SID via the Global Route Manager (GRM).
  - Optionally configure a route map to filter routes that are imported or exported between the VRF and I-SID defined in an L3 VPN binding.
  - Optionally configure additional methods for route leaking, such as route redistribution to allow routing between a VRF instance and an I-SID or between two I-SIDs.

IP over SPB Configuration Guidelines

Consider the following guidelines when configuring IP over SPB:

- An L3 VPN interface serves as an IP gateway to access remote networks. The network administrator must ensure the IP subnet reachability of the L3 VPN addresses on the same SPBM I-SID.
- IPv4 L3 VPN interfaces use dynamic ARP and IPv6 L3 VPN interfaces use neighbor discovery to learn the MAC addresses of other L3 VPN interfaces and provide next-hop forwarding information to the switch.
- The following SPB inline routing (L3 VPN) solutions are available. Each solution is supported on specific switch platforms, as noted in the table on page 7-19.
  - Single-pass inline routing—an L3 VPN interface is defined through the configuration of an IP interface that is bound to an SPB service.
  - Two-pass inline routing using a single front panel port—an L3 VPN interface is defined through the configuration of front panel ports or a link aggregate to run in loopback mode.
- When creating an L3 VPN interface for single-pass inline routing, assigning an existing SPB service to an IP interface is required. The IP address assigned to the service interface is used as a gateway address to bind a VRF instance to an SPB backbone service instance identifier (I-SID). When assigning an SPB service to an IP interface, consider the following:
  - VLAN translation is implicitly enabled when a service is assigned to an IP interface regardless of whether or not VLAN translation is enabled for the service; the VLAN translation status is no longer configurable as long as the service is bound to an IP interface.
  - Configuring multiple IP addresses (IP multinetting) for the same service-based interface is not supported.

- When creating an L3 VPN interface for two-pass inline routing, consider the following:
  - The loopback function is defined on a single port or link aggregate (instead of two separate ports or link aggregates). This eliminates the need for a physical cable to connect two different ports to create the loopback. The same port is assigned to the L3 VPN VLAN and is also configured as a service access port.
  - A dedicated VLAN and a port or link aggregate configured to operate in the loopback mode is required.
  - Once a port or link aggregate is configured to run in the loopback mode, no other functionality is supported on the port or link aggregate.
  - The dedicated VLAN is reserved for the L3 VPN and can only be associated with the loopback port or link aggregate. To ensure this, configure the L3 VPN IP interface using the router port and tagged VLAN options, where the router port is the loopback port and the tagged VLAN is the dedicated VLAN.
  - The full bandwidth of the loopback port is available to forward traffic in and out of the SPB service. In addition, multiple front-panel loopback ports can be combined into a static loopback link aggregate for redundancy and to increase bandwidth.
  - When configuring the loopback port or link aggregate as an access port and creating an SPB service, enable VLAN translation.
  - Once the loopback mode is enabled for a link aggregate, the link aggregate is dedicated to providing loopback functionality for an SPB L3 VPN inline routing configuration. The loopback mode is disabled only when the link aggregate is deleted.
- There are two scenarios for mapping a VRF instance to an I-SID:
  - One VRF to one I-SID
  - One VRF to many I-SIDs
- When an L3 VPN (ISIS-SPB) “bind” entry goes active, ISIS-SPB will export learned routes from the SPB network to the GRM, which triggers the GRM to send IP routes from the corresponding VRF to ISIS-SPB using the ISID and gateway IP address as the next hop.
- The difference between an L3 VPN (ISIS-SPB) bind entry and a redistribution entry is as follows:
  - A bind entry binds only one I-SID to one VRF, IP gateway instance. IP VPN routes are then imported and exported bidirectionally between the VRF and I-SID.
  - A redistribution entry allows multiple VRFs to redistribute routes into one I-SID. However, IP VPN routes are only redistributed into the I-SID; there is no bidirectional exchange of routes between the VRF and I-SID. Redistribution is mainly used when routing between I-SIDs is required.
- If exchanging routes between different VRF networks associated with different I-SIDs (inter-service routing) is required, the L3 VPN (ISIS-SPB) approach supports the following route leaking methods:
  - VRF-to-VRF (route import/export)
  - VRF-to-I-SID (route redistribution)
  - I-SID-to-I-SID (route redistribution)
  - I-SID-to-VRF (route import)

Configuring the L3 VPN Interface

Identify the BEBs that will participate in routing L3 traffic through the SPBM core. On each of these BEBs, configure the required L3 VPN interface configuration. Use one of the following options to configure this type of interface based on the switch platform:

- L3 VPN Interface: External Loopback.
- L3 VPN Interface: Inline Routing (Service-Based IP Interface).
- L3 VPN Interface: Inline Routing with a Front-Panel Port.
- L3 VPN Interface: Inline Routing with a Static Link Aggregate.

L3 VPN Interface: External Loopback

The following commands create an external loopback port configuration that will serve as an L3 VPN interface:

```
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
-> spb bvlan 500
-> spb isis control-bvlan 500
-> service access port 1/1/2 vxlan-xlation
-> service 10 spb isid 1000 bvlan 500 admin-state enable
-> service 10 sap port 1/1/2:400
-> vrf create vrf-1
vrf-1::-> ip interface IPv4-L3vpn1 vlan 400 address 10.1.1.1/24
vrf-1::-> ipv6 interface IPv6-L3vpn1 vlan 400 address 1000::1
vrf-1::-> ipv6 interface IPv6-L3vpn2 vlan 400 address 2000::1
```

Once the above loopback port configuration is defined, use a physical cable to connect port 1/1/1 to access port 1/1/2. The SPB BVLAN and I-SID are required to create the SAP for access port 1/1/2. See "IPv4 L3 VPN External Loopback and Inline Routing: Two I-SIDs, One VRF" on page 7-89 for an example configuration of an L3 VPN loopback interface.

L3 VPN Interface: Inline Routing (Service-Based IP Interface)

The following commands provide an example for creating an L3 VPN service-based IP interface for single-pass inline routing:

```
-> spb bvlan 500
-> spb isis control-bvlan 500
-> service 10 spb isid 1000 bvlan 500 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface IPv4-L3vpn3 service 10 address 100.1.1.1/8
vrf-1::-> ipv6 interface IPv6-L3vpn4 service 10 address 4000::1
```

In this example, the "IPv4-L3vpn3" and "IPv6-L3vpn4" interfaces are created in the "vrf-1" instance and bound to SPB service 10. See "IPv4 L3 VPN Inline Routing: Service-Based (Two I-SIDS, Two VRFs)" on page 7-82 for an example configuration of an L3 VPN service-based interface.

Note. If inline routing is *not* going to be configured on the OmniSwitch 6860N or on the OmniSwitch 6900-X48C6/X48C4E/T48C6/V48C8, inline routing support can be disabled on these models using the **service l3vpn profile** command. This will free up additional switch resources for other non-inline routing functionality. See the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the **service l3vpn profile** command.

L3 VPN Interface: Inline Routing with a Front-Panel Port

The following commands provide an example for creating an L3 VPN interface using a front-panel loopback port:

1 Use the **interfaces** command with the **loopback** option to configure a port to operate in the loopback mode. For example:

```
-> interfaces port 1/1/18 loopback
```

2 Use the **service access** command to configure the loopback port as a service access port with VLAN translation enabled. For example:

```
-> service access port 1/1/18 vlan-xlation
```

3 Use the **service spb** command to create an SPB service with VLAN translation enabled. For example;

```
-> service 10 spb isid 1000 bvlan 500 vlan-xlation
```

4 Use the **service sap** command to create a SAP that binds the loopback port to the SPB service and the L3 VPN VLAN. For example:

```
-> service 10 sap port 1/1/18:400
```

5 Use the **vrf** command to create the VRF instance (or use the default VRF instance) in which the IP VPN interface will be created. For example:

```
-> vrf create vrf-1
```

6 To create an IPv4 VPN interface, use the **ip interface rtr-port** command with the **tagged vlan** option. Specify the VLAN ID of the L3 VPN VLAN with this command. For example:

```
vrf-1::-> ip interface L3VPN address 100.1.1.1/8 rtr-port port 1/1/18 tagged
vlan 400
```

7 To create an IPv6 VPN interface, use the **ipv6 interface rtr-port** command with the **vlan** option and the **ipv6 address** command. Specify the VLAN ID of the L3 VPN VLAN with this command. For example:

```
vrf-1::-> ipv6 interface L3VPN rtr-port port 1/1/18 tagged vlan 400
vrf-1::-> ipv6 address 2001:db8:10::1/64 L3VPN
```

See "IPv4 L3 VPN Inline Routing: Front-Panel Ports" on page 7-86 for an example configuration in which a front-panel port serves as an L3 VPN interface.

L3 VPN Interface: Inline Routing with a Static Link Aggregate

The following commands provide an example for creating an L3 VPN interface using a static loopback link aggregate comprised of loopback ports:

1 Use the **interfaces loopback** command with the **loopback** option to configure ports to operate in the loopback mode. For example:

```
-> interfaces port 1/1/18-21 loopback
```

2 Use the **linkagg static agg loopback** command to configure a static link aggregate to operate in the loopback mode. For example:

```
-> linkagg static agg 1 size 4
-> linkagg static agg 1 loopback
```

3 Use the **linkagg static port agg** command to assign the loopback ports to the link aggregate. For example:

```
-> linkagg static port 1/1/18-21 agg 1
```

4 Use the **service access** command to configure the loopback link aggregate as a service access port with VLAN translation enabled. For example:

```
-> service access linkagg 1 vlan-xlation
```

5 Use the **service spb** command to create an SPB service with VLAN translation enabled. For example;

```
-> service 10 spb isid 1000 bvlan 500 vlan-xlation
```

6 Use the **service sap** command to create a SAP that binds the loopback link aggregate to the SPB service. For example:

```
-> service 10 sap linkagg 10:400
```

7 Create the VRF instance (or use the default VRF instance) that will be bound to the SPB service instance and IP gateway using the **vrf** command. For example:

```
-> vrf create vrf-1
```

8 To create an IPv4 VPN interface, use the **ip interface rtr-port** command with the **vlan** option. For example:

```
vrf-1::-> ip interface L3VPN address 100.1.1.1/8 rtr-port linkagg 10 tagged vlan
400
```

9 To create an IPv6 VPN interface, use the **ipv6 interface rtr-port** command with the **vlan** option and the **ipv6 address** command. For example:

```
vrf-1::-> ipv6 interface L3VPN rtr-port linkagg 10 tagged vlan 400
vrf-1::-> ipv6 address 2001:db8:10::1/64 L3VPN
```

See "IPv4 L3 VPN Inline Routing: Front-Panel Ports" on page 7-86 for an example configuration in which a static link aggregate comprised of front-panel loopback ports serves as an L3 VPN interface.

Configuring the VPN-Lite Solution

The VPN-Lite approach requires configuring routing protocols on the L3 VPN interface that will exchange routes between the VLAN domain (VRF instance) and the SPB service domain (I-SID). For example, the following commands configure static routes for L3 VPN interfaces (gateway address):

```
vrf-1::-> ip static-route 20.0.0.0/24 gateway 10.1.1.1
vrf-1::-> ip static-route 10.0.0.0/24 gateway 20.1.1.1

vrf-1::-> ipv6 static-route 1500::/16 gateway 1000::1
vrf-1::-> ipv6 static-route 1600::/16 gateway 2000::1
```

Configuring the L3 VPN (ISIS-SPB) Solution

The L3 VPN (ISIS-SPB) approach requires configuring a VRF-ISID binding to identify the L3 VPN interface that will exchange routes between the VLAN domain (VRF instance) and the SPB service domain (I-SID). VRF import and export commands are used to exchange routes between the VRF and the I-SID specified in the binding configuration. For example:

```
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 10.1.1.1 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
```

In this example, “vrf-1” is bound to SPB I-SID 1000 and gateway 10.1.1.1 identifies the IPv4 L3 VPN interface. All IPv4 routes in “vrf-1” are exported to the Global Route Manager (GRM), which then exports the routes to I-SID 1000. The last command in this sequence sets up the import of I-SID 1000 routes from the GRM into the “vrf-1” instance. The **all-routes** parameter specifies that no route-map filtering is applied to exported or imported routes; all routes are allowed.

```
vrf-1::-> ipv6 export all-routes
vrf-1::-> vrf default
-> spb ipvpn6 bind vrf-1 isid 3000 gateway 1000::1
-> vrf vrf-1
vrf-1::-> ipv6 import isid 3000 all-routes
```

In this example, “vrf-1” is bound to SPB I-SID 3000 and gateway 1000::1 identifies the IPv6 L3 VPN interface. All IPv6 routes in “vrf-1” are exported to the GRM, which then exports the routes to I-SID 3000. The last command in this sequence sets up the import of I-SID 3000 routes from the GRM into the “vrf-1” instance. The **all-routes** parameter specifies that no route-map filtering is applied to exported or imported routes; all routes are allowed.

Filtering Imported or Exported Routes

To filter routes that are imported or exported, define a route map to use with the **ip import** or **ip export** commands. For example, the following commands create the “ipvpn-vrf1” route map and filter exported and imported routes:

```
vrf-1::-> ip access-list ipaddr
vrf-1::-> ip access-list ipaddr address 15.0.0.0/8 action permit redist-control
all-subnets
vrf-1::-> ip route-map ipvpn-vrf1 sequence-number 1 action permit
vrf-1::-> ip route-map ipvpn-vrf1 sequence-number 1 match ip-address ipaddr
vrf-1::-> ip export ipvpn-vrf1
vrf-1::-> ip import isid 1000 ipvpn-vrf1

vrf-1::-> ipv6 access-list ip6addr
vrf-1::-> ipv6 access-list ip6addr address 2001::/64 action permit redist-
control all subnets
vrf-1::-> ip route-map ipvpn6-vrf1 sequence-number 1 action permit
vrf-1::-> ip route-map ipvpn6-vrf1 sequence-number 1 match ipv6-address ip6addr
vrf-1::-> ipv6 export ipvpn6-vrf1
vrf-1::-> ipv6 import isid 3000 ipvpn6-vrf1
```

Configuring Route Redistribution Between VRFs and/or I-SIDs

The L3 VPN (ISIS-SPB) solution also allows for the redistribution of routes between a VRF instance and an I-SID or between two I-SIDs (inter-I-SID route leaking). For example, the following commands

redistribute routes from I-SID 2000 into I-SID 1000, from “vrf-1” into I-SID 2000, from I-SID 4000 into I-SID 3000, and from “vrf-1” into I-SID 4000.

```
-> spb ipvpn redist source-isid 2000 destination-isid 1000 all-routes
-> spb ipvpn redist source-vrf vrf-1 destination-isid 2000 all-routes
-> spb ipvpn6 redist source-isid 4000 destination-isid 3000 all-routes
-> spb ipvpn6 redist source-vrf vrf-1 destination-isid 4000 all-routes
```

See “IPv4 L3 VPN External Loopback Interfaces: I-SID Routing in One VRF” on page 7-91 and “IPv4 L3 VPN External Loopback Interfaces: I-SID Routing in Two VRFs” on page 7-95 for examples of using route redistribution in an IP over SPB configuration.

Verifying L3 VPN Configuration and Routes

VRFs are bound to I-SIDs to identify a VRF mapping to a specific SPB service instance for the purposes of exchanging routes between the VRF and I-SID via the switch GRM. To verify the VRF mapping configuration on the local switch, use the **show spb ipvpn bind** or **show spb ipvpn6 bind** command. For example:

```
-> show spb ipvpn bind
Legend: * indicates bind entry is active
SPB IPVPN Bind Table:
       VRF              ISID            Gateway                 Route-Map
--------------------+------------+----------------------+----------------------
* vrf-1                  1000         10.1.1.1
* vrf-2                  2000         20.1.1.1

Total Bind Entries: 2

-> show spb ipvpn6 bind
Legend: * indicates bind entry is active
SPB IPVPN Bind Table:
       VRF              ISID            Gateway                 Route-Map
--------------------+------------+----------------------+----------------------
* ospf                   3000         1000::1
* ospf1                  4000         2000::1

Total Bind Entries: 2
```

In addition to exchanging routes between VRFs and I-SIDs, it is also possible to configure redistribution of routes between two I-SIDs or between a VRF and an I-SID. To verify the redistribution configuration for L3 VPN routes, use the **show spb ipvpn redist** or **show spb ipvpn6 redist** command. For example:

```
-> show spb ipvpn redist
Legend: * indicates redist entry is active
SPB IPVPN Redist ISID Table:
  Source-ISID           Destination-ISID     Route-Map
----------------------+--------------------+--------------------
* 2000                  1000
* 2001                  1001

Total Redist ISID Entries: 2

Legend: * indicates redist entry is active
SPB IPVPN Redist VRF Table:
  Source-VRF            Destination-ISID     Route-Map
----------------------+--------------------+--------------------
* vrf-1                 2000

Total Redist Vrf Entries: 0
```

```
-> show spb ipvpnv6 redist
Legend: * indicates redist entry is active
SPB IPVPN6 Redist ISID Table:
  Source-ISID           Destination-ISID     Route-Map
----------------------+--------------------+--------------------
* 4000                  3000
* 4001                  3001

Total Redist ISID Entries: 2

Legend: * indicates redist entry is active
SPB IPVPN6 Redist VRF Table:
  Source-VRF            Destination-ISID     Route-Map
----------------------+--------------------+--------------------
* vrf-1                 4000

Total Redist Vrf Entries: 0
```

To display the L3 VPN route table, use the **show spb ipvpn route-table** or **show spb ipvpn6 route-table** command. For example:

```
-> show spb ipvpn route-table
Legend: * indicates IPVPN route has matching locally configured ISID
SPB IPVPN Route Table:
                                                  Source Bridge
     ISID  Destination         Gateway            (Name : BMAC)                Metric
---------+--------------------+-----------------+---------------------------+------
*    4001  1.1.1.0/24           1.1.1.1           L2-DUT1 : 00:e0:b1:db:c3:65     1
*    4001  1.1.1.0/24           1.1.1.2           L2-DEV1 : e8:e7:32:00:23:f9     1
*    4001  2.2.2.0/24           1.1.1.2           L2-DEV1 : e8:e7:32:00:23:f9     1
*    4001  10.10.10.0/24        1.1.1.1           L2-DUT1 : 00:e0:b1:db:c3:65     1
*    4003  1.1.1.0/24           2.2.2.2           L2-DEV1 : e8:e7:32:00:23:f9     1
*    4003  2.2.2.0/24           2.2.2.1           L2-DUT2 : 00:e0:b1:dd:99:db     1
*    4003  2.2.2.0/24           2.2.2.2           L2-DEV1 : e8:e7:32:00:23:f9     1
*    4003  10.10.10.0/24        2.2.2.2           L2-DEV1 : e8:e7:32:00:23:f9     1

Routes: 8

-> show spb ipvpn6 route-table
Legend: * indicates IPVPN6 route has matching locally configured ISID
SPB IPVPN6 Route Table:
                                                  Source Bridge
     ISID  Destination         Gateway            (Name : BMAC)                Metric
---------+--------------------+-----------------+---------------------------+------
*    4001  1501:0001::/32       1000::1           L2-DUT1 : 00:e0:b1:db:c3:65     1
*    4001  1601:0001::/32       1000::2           L2-DEV1 : e8:e7:32:00:23:f9     1
*    4001  1701:0001::/32       1000::2           L2-DEV1 : e8:e7:32:00:23:f9     1
*    4001  1901:0001::/32       1000::1           L2-DUT1 : 00:e0:b1:db:c3:65     1
*    4003  1501:0001::/32       2000::2           L2-DEV1 : e8:e7:32:00:23:f9     1
*    4003  1601:0001::/32       2000::1           L2-DUT2 : 00:e0:b1:dd:99:db     1
*    4003  1701:0001::/32       2000::2           L2-DEV1 : e8:e7:32:00:23:f9     1
*    4003  1901:0001::/32       2000::2           L2-DEV1 : e8:e7:32:00:23:f9     1

Routes: 8
```

IP over SPB Configuration Examples

This section contains diagrams and CLI command examples for configuring the following IP over SPB scenarios:

- "IPv4 L3 VPN Inline Routing: Service-Based (Two I-SIDS, Two VRFs)" on page 7-82.
- "IPv4 L3 VPN Inline Routing: Front-Panel Ports" on page 7-86.
- "IPv4 L3 VPN External Loopback and Inline Routing: Two I-SIDs, One VRF" on page 7-89.
- "IPv4 L3 VPN External Loopback Interfaces: I-SID Routing in One VRF" on page 7-91.
- "IPv6 L3 VPN External Loopback Interfaces: I-SID Routing in One VRF" on page 7-93.
- "IPv4 L3 VPN External Loopback Interfaces: I-SID Routing in Two VRFs" on page 7-95.
- "IPv6 L3 VPN External Loopback Interfaces: I-SID Routing in Two VRFs" on page 7-97.
- "L3 VPN External Loopback Migration to Inline Routing" on page 7-99.

IPv4 L3 VPN Inline Routing: Service-Based (Two I-SIDS, Two VRFs)

In this sample IPv4 over SPB topology, Customer Edge (CE) devices can communicate with other CE devices through L3 VPN services that traverse the SPB backbone network. Because the OmniSwitch 9900 is used in this example topology, the IPv4 L3 VPN interfaces are defined by creating an IPv4 interface within each VRF instance that is bound to an SPB service.

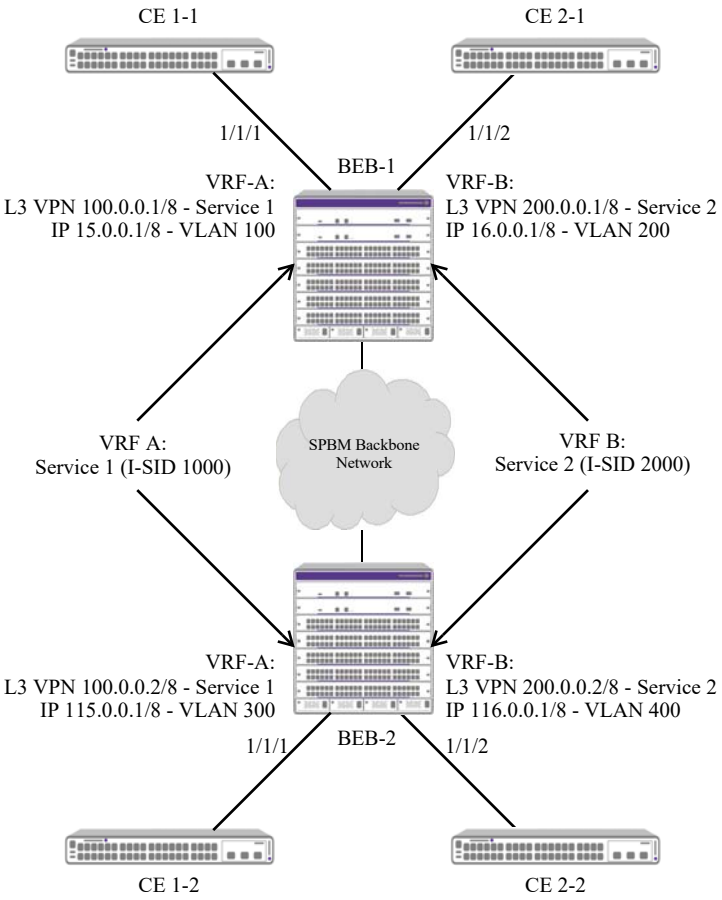


Figure 7-17 : IPv4 L3 VPN Service-based Interfaces (Inline Routing)

In this topology, Customer 1 requires L3 VPN services between CE 1-1 and CE 1-2, and Customer 2 requires L3 VPN services between CE 2-1 and CE 2-2.

- BEB-1 is connected to two customer edge devices (CE 1-1 and CE 2-1).
- BEB-2 is connected to two customer edge devices (CE 1-2 and CE 2-2).
- SPB service 1/I-SID 1000 and VRF A are dedicated to Customer 1.
- SPB service 2/I-SID 2000 and VRF B are dedicated to Customer 2.
- An IPv4 interface is configured in VRF A and bound to SPB service 1 to define an IPv4 L3 VPN interface for Customer 1.
- An IPv4 interface is configured in VRF B and bound to SPB service 2 to define an IPv4 L3 VPN interface for Customer 2.
- VPN routes are learned on the BEB switches through L3 VPN (ISIS-SPB) or VPN-lite and are then installed into the appropriate VRFs.

- The nexthop interface for VPN routes is through the IPv4 L3 VPN interfaces. For example,
  - The VPN routes in BEB-1 for VRF-A are installed with the next hop set to100.0.0.2, and the VPN routes in BEB-2 for VRF-A are installed with the nexthop set to100.0.0.1.
  - The VPN routes in BEB-1 for VRF-B are installed with the next hop set to 200.0.0.2, and the VPN routes in BEB-2 for VRF-B are installed with the next hop set to 200.0.0.1.

The following CLI command examples are used to configure the sample IPv4 over SPB topology shown in "IPv4 L3 VPN Service-based Interfaces (Inline Routing)" on page 7-82.

BEB-1:

```
-> vlan 100
-> vlan 100 members port 1/1/1 tagged
-> service 1 spb isid 1000 bvlan 40 admin-state enable
-> vrf create vrf-A
vrf-A::-> ip interface ip-a1 address 15.0.0.1/8 vlan 100
vrf-A::-> ip interface l3vpn-a1 address 100.0.0.1/8 service 1
vrf-A::-> vrf default

-> vlan 200
-> vlan 200 members port 1/1/2 tagged
-> service 2 spb isid 2000 bvlan 41 admin-state enable
-> vrf create vrf-B
vrf-B::-> ip interface ip-b1 address 16.0.0.1/8 vlan 200
vrf-B::-> ip interface l3vpn-b1 address 200.0.0.1/8 service 2
```

BEB-2:

```
-> vlan 300
-> vlan 300 members port 1/1/1 tagged
-> service 1 spb isid 1000 bvlan 40 admin-state enable
-> vrf create vrf-A
vrf-A::-> ip interface ip-a2 address 115.0.0.1/8 vlan 300
vrf-A::-> ip interface l3vpn-a2 address 100.0.0.2/8 service 1

-> vlan 400
-> vlan 400 members port 1/1/2 tagged
-> service 2 spb isid 2000 bvlan 41 admin-state enable
-> vrf create vrf-B
vrf-B::-> ip interface ip-b2 address 116.0.0.1/8 vlan 400
vrf-B::-> ip interface l3vpn-b2 address 200.0.0.2/8 service 2
```

VPN-Lite

The VPN-Lite approach requires configuring routing protocols to run on the L3 VPN interface or defining static routes with a remote L3 VPN interface as a gateway. The following commands provide examples of configuring either OSPF or static routes for a VPN-Lite configuration.

BEB-1 (OSPF—the interface name specified is the name of the IPv4 L3 VPN service-based interface):

```
vrf-A::-> ip load ospf
vrf-A::-> ip ospf interface l3vpn-a1
vrf-A::-> ip ospf area 0.0.0.0
vrf-A::-> ip ospf interface l3vpn-a1 area 0.0.0.0
vrf-A::-> ip ospf interface l3vpn-a1 admin-state enable
vrf-A::-> ip ospf admin-state enable

vrf-B::-> ip load ospf
vrf-B::-> ip ospf interface l3vpn-b1
vrf-B::-> ip ospf area 0.0.0.0
vrf-B::-> ip ospf interface l3vpn-b1 area 0.0.0.0
```

```
vrf-B::-> ip ospf interface l3vpn-b1 admin-state enable
vrf-B::-> ip ospf admin-state enable
```

BEB-1 (Static Routes—the gateway IP address specified is the IP address of the IPv4 L3 VPN service-based interface):

```
vrf-A::-> ip static-route 115.0.0.0/8 gateway 100.0.0.2
vrf-B::-> ip static-route 116.0.0.0/8 gateway 200.0.0.2
```

BEB-2 (OSPF—the interface name specified is the name of the IPv4 L3 VPN service-based interface):

```
vrf-A::-> ip load ospf
vrf-A::-> ip ospf interface l3vpn-a2
vrf-A::-> ip ospf area 0.0.0.0
vrf-A::-> ip ospf interface l3vpn-a2 area 0.0.0.0
vrf-A::-> ip ospf interface l3vpn-a2 admin-state enable
vrf-A::-> ip ospf admin-state enable

vrf-B::-> ip load ospf
vrf-B::-> ip ospf interface l3vpn-b2
vrf-B::-> ip ospf area 0.0.0.0
vrf-B::-> ip ospf interface l3vpn-b2 area 0.0.0.0
vrf-B::-> ip ospf interface l3vpn-b2 admin-state enable
vrf-B::-> ip ospf admin-state enable
```

BEB-2 (Static Routes—the gateway IP address specified is the IP address of the IPv4 L3 VPN service-based interface):

```
vrf-A::-> ip static-route 15.0.0.0/8 gateway 100.0.0.1
vrf-B::-> ip static-route 16.0.0.0/8 gateway 200.0.0.1
```

L3 VPN

The L3 VPN (ISIS-SPB) approach exchanges L3 routes between VRFs. Instead of configuring routing protocols on the L3 VPN interface (VPN-Lite), VRFs are bound to backbone I-SIDs to connect VRFs across the SPBM network. IP routes are then imported into ISIS-SPB from the VRFs where IPVPN TLVs are used to carry the routes through the network to other SPBM BEB switches.

The following commands are used only when configuring an L3 VPN (ISIS-SPB) configuration (the gateway IP address specified is the IP address of the IPv4 L3 VPN service-based interface).

BEB-1:

```
vrf-A::-> ip export all-routes
vrf-A::-> vrf default
-> spb ipvpn bind vrf-A isid 1000 gateway 100.0.0.1 all-routes
-> vrf vrf-A
vrf-A::-> ip import isid 1000 all-routes
vrf A::-> vrf default

-> vrf vrf-B
vrf-B::-> ip export all-routes
vrf-B::-> vrf default
-> spb ipvpn bind vrf-B isid 2000 gateway 200.0.0.1 all-routes
-> vrf vrf-B
vrf-B::-> ip import isid 2000 all-routes
vrf B::-> vrf default
```

BEB-2:

```
vrf-A::-> ip export all-routes
vrf-A::-> vrf default
```

```
-> spb ipvpn bind vrf-A isid 1000 gateway 100.0.0.2 all-routes
-> vrf vrf-A
vrf-A::-> ip import isid 1000 all-routes
vrf A::-> vrf default

-> vrf vrf-B
vrf-B::-> ip export all-routes
vrf-B::-> vrf default
-> spb ipvpn bind vrf-B isid 2000 gateway 200.0.0.2 all-routes
-> vrf vrf-B
vrf-B::-> ip import isid 2000 all-routes
vrf B::-> vrf default
```

IPv4 L3 VPN Inline Routing: Front-Panel Ports

In this sample IP over SPB topology, routing between Customer Edge (CE) devices across the SPB backbone network is achieved through the L3 VPN interface configured on BEB-A and BEB-B. Because the OmniSwitch 6900-V72 is used in this example topology, the L3 VPN interfaces are defined by using front-panel ports configured to operate in the loopback mode.

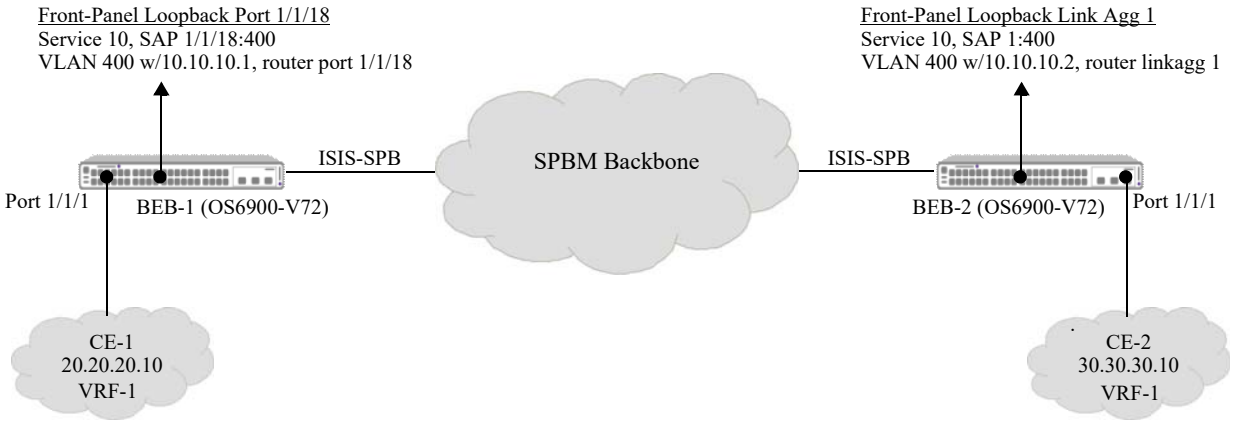


Figure 7-18 : L3 VPN Front-Panel Ports (Inline Routing)

In this topology, L3 VPN services provide routing between CE devices.

- CE devices are connected to BEB-1 and BEB-2 in IP subnet 20.20.20.0/24 and 30.30.30.0/24, respectively.
- The CE devices can either be connected to bridge ports or access ports on the BEB switches. The CLI commands included in this sample topology provide examples for both scenarios (CE connected to bridge port; CE connected to access port). When a CE device is connected to an access port, the L3 VPN subnet can be extended to the device.
- An L3 VPN interface is configured on BEB-1 and BEB-2 in IP subnet 10.10.10.0/24.
- The L3 VPN interface on BEB-1 is comprised of a front-panel loopback port that is bound to an IP interface (10.10.10.1) and VLAN 400. The same port is also configured as a service access port that is used to define a Service Access Point (SAP) binding with service 10.
- The L3 VPN interface on BEB-2 is comprised of a loopback link aggregate to which multiple front-panel loopback ports are assigned. The link aggregate is bound to an IP interface (10.10.10.2) and VLAN 400. The same link aggregate is also configured as a service access link aggregate that is used to define a SAP binding with service 10.
- Once the L3 VPN interfaces are defined on each BEB, then either the VPN-Lite or L3 VPN (ISIS-SPB) method is configured to exchange routes across the SPB network. The CLI commands included in this sample topology provide examples of how to configure both methods.
  - VPN-Lite requires routing protocols to be configured on all the IP interfaces. See "VPN-Lite" on page 7-88.
  - L3 VPN (ISIS-SPB) requires binding the L3 VPN interface to an SPB I-SID to exchange all routes in the associated VRF instance with other SPB bridges. See "L3 VPN (ISIS-SPB)" on page 7-88.

The following CLI command examples are used to configure the sample IP over SPB topology shown in "L3 VPN Front-Panel Ports (Inline Routing)" on page 7-86.

BEB-1 (CE devices connected to bridge ports):

```
-> interface port 1/1/18 loopback

-> vlan 20
-> vlan 20 member port 1/1/1 tagged

-> ip interface IPVPN address 10.10.10.1/24 rtr-port port 1/1/18 tagged vlan 400
-> ip interface IP-CE address 20.20.20.1/24 vlan 20

-> service access port 1/1/18 vlan-xlation enable
-> service spb 10 isid 1000 bvlan 4000 vlan-xlation enable
-> service 10 sap port 1/1/18:400
```

BEB-1 (CE devices connected to access ports):

```
-> interface port 1/1/18 loopback

-> ip interface IPVPN address 10.10.10.1/24 rtr-port port 1/1/18 tagged vlan 400

-> service access port 1/1/18 vlan-xlation enable
-> service spb 10 isid 1000 bvlan 4000 vlan-xlation enable
-> service 10 sap port 1/1/18:400
-> service 10 sap port 1/1/1:400
```

BEB-2 (CE devices connected to bridge ports):

```
-> interface port 1/1/18 loopback
-> interface port 2/1/18 loopback
-> linkagg static agg 1 size 4 loopback
-> linkagg static port 1/1/18 agg 1
-> linkagg static port 2/1/18 agg 1

-> vlan 30
-> vlan 30 member port 1/1/1 tagged

-> ip interface IPVPN address 10.10.10.2/24 rtr-port linkagg 1 tagged vlan 400
-> ip interface IP-CE address 30.30.30.2/24 vlan 30

-> service access linkagg 1 vlan-xlation enable
-> service spb 10 isid 1000 bvlan 4000 vlan-xlation enable
-> service 10 sap linkagg 1:400
```

BEB-2 (CE devices connected to access ports):

```
-> interface port 1/1/18 loopback
-> interface port 2/1/18 loopback
-> linkagg static agg 1 size 4 loopback
-> linkagg static port 1/1/18 agg 1
-> linkagg static port 2/1/18 agg 1

-> ip interface IPVPN address 10.10.10.2/24 rtr-port linkagg 1 tagged vlan 400

-> service access linkagg 1 vlan-xlation enable
-> service spb 10 isid 1000 bvlan 4000 vlan-xlation enable
-> service 10 sap linkagg 1:400
-> service 10 sap port 1/1/1:400
```

VPN-Lite

The VPN-Lite approach requires configuring routing protocols to run on all the interfaces or defining static routes with a remote interface as a gateway. The following commands provide examples of configuring either OSPF or static routes for a VPN-Lite configuration.

BEB-1 (OSPF—the interface name specified is the name of the L3 VPN interface and the CE interface):

```
-> ip load ospf
-> ip ospf area 0.0.0.0
-> ip ospf interface "IPVPN"
-> ip ospf interface "IPVPN" area 0.0.0.0
-> ip ospf interface "IPVPN" admin-state enable
-> ip ospf interface "IP-CE"
-> ip ospf interface "IP-CE" area 0.0.0.0
-> ip ospf interface "IP-CE" admin-state enable
-> ip ospf admin-state enable
```

BEB-1 (Static Routes—the gateway IP address specified is the IP address of the L3 VPN interface):

```
-> ip static-route 20.20.20.0/24 gateway 10.10.10.1
```

BEB-2 (OSPF—the interface name specified is the name of the L3 VPN interface and the CE interface):

```
-> ip load ospf
-> ip ospf area 0.0.0.0
-> ip ospf interface "IPVPN"
-> ip ospf interface "IPVPN" area 0.0.0.0
-> ip ospf interface "IPVPN" admin-state enable
-> ip ospf interface "IP-CE"
-> ip ospf interface "IP-CE" area 0.0.0.0
-> ip ospf interface "IP-CE" admin-state enable
-> ip ospf admin-state enable
```

BEB-2 (Static Routes—the gateway IP address specified is the IP address of the IPv4 L3 VPN service-based interface):

```
-> ip static-route 30.30.30.0/24 gateway 10.10.10.2
```

L3 VPN (ISIS-SPB)

The L3 VPN (ISIS-SPB) approach exchanges L3 routes between VRFs. Instead of configuring routing protocols on the L3 VPN interface (VPN-Lite), VRFs are bound to backbone I-SIDs to connect VRFs across the SPBM network. IP routes are then imported into ISIS-SPB from the VRFs where IPVPN TLVs are used to carry the routes through the network to other SPBM BEB switches.

The following commands are used only when configuring an L3 VPN (ISIS-SPB) configuration (the gateway IP address specified is the IP address of the L3 VPN interface).

BEB-1:

```
-> ip export all-routes
-> spb ipvpn bind vrf default isid 1000 gateway 10.10.10.1 all-routes
-> ip import isid 1000 all-routes
```

BEB-2:

```
-> ip export all-routes
-> spb ipvpn bind vrf default isid 1000 gateway 10.10.10.2 all-routes
-> ip import isid 1000 all-routes
```

IPv4 L3 VPN External Loopback and Inline Routing: Two I-SIDs, One VRF

In this sample IPv4 over SPB topology, Network A can communicate with Network B across two I-SIDs. This scenario could be expanded to connect multiple customer sites together to form a VPN cloud.

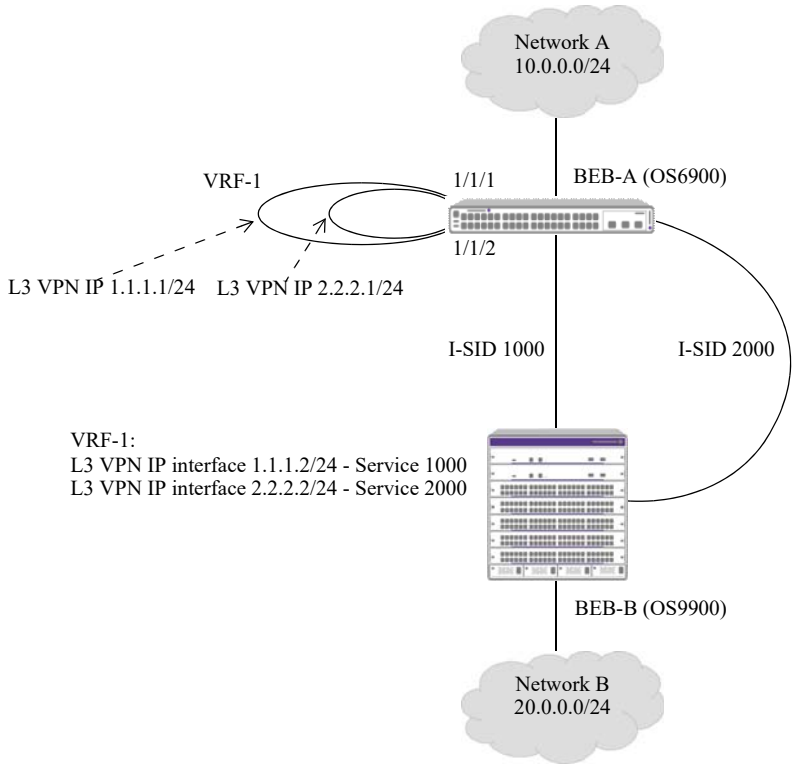


Figure 7-19 : IPv4 L3 VPN External Loopback and Service-based Interfaces

In this topology,

- A physical loopback port configuration is required on BEB-A (OS6900). Port 1/1/1 is the L3 VPN router port, port 1/1/2 is the L3 VPN access port, and VLAN 200 and VLAN 400 are the L3 VPN VLANs.
  - Port 1/1/1 is tagged with VLANs 200 and 400.
  - Access port 1/1/2 is assigned to SAPs that are each associated with an I-SID (an SPB service instance) that will forward VLAN 200 and 400 traffic through the SPB backbone network.
  - A physical cable is connected to port 1/1/1 and to port 1/1/2 to create the connection between the VLAN and service domains.
  - An IPv4 L3 VPN interface is configured on VLAN 200 (1.1.1.1/24) and VLAN 400 (2.2.2.1/24).
- A physical loopback port configuration is not used on BEB-B (OS9900). Instead, an IPv4 L3 VPN interface is configured on service 1000 (1.1.1.2/24) and service 2000 (2.2.2.2/24).
- On BEB-A and BEB-B, the VPN-Lite or L3 VPN solution is configured.

The following CLI command examples are used to configure the sample IPv4 over SPB topology shown in "IPv4 L3 VPN External Loopback and Service-based Interfaces" on page 7-89.

BEB-A:

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> vlan 400
```

```
-> vlan 400 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 vlan 200 address 1.1.1.1/24
vrf-1::-> ip interface l3vpn2 vlan 400 address 2.2.2.1/24
```

BEB-B (OS9900):

```
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 service 1000 address 1.1.1.2/24
vrf-1::-> ip interface l3vpn2 service 2000 address 2.2.2.2/24
```

VPN-Lite

The VPN-Lite approach requires configuring routing protocols to run on the L3 VPN interface or defining static routes with a remote L3 VPN interface as a gateway. The following commands provide examples of configuring static routes for a VPN-Lite configuration.

BEB-A:

```
vrf-1::-> ip static-route 20.0.0.0/24 gateway 1.1.1.2
vrf-1::-> ip static-route 20.0.0.0/24 gateway 2.2.2.2
```

BEB-B (OS9900):

```
vrf-1::-> ip static-route 10.0.0.0/24 gateway 1.1.1.1
vrf-1::-> ip static-route 10.0.0.0/24 gateway 2.2.2.1
```

L3 VPN

The L3 VPN (ISIS-SPB) approach exchanges L3 routes between VRFs. Instead of configuring routing protocols on the L3 VPN interface (VPN-Lite), VRFs are bound to backbone I-SIDs to connect VRFs across the SPBM network. IP routes are then imported into ISIS-SPB from the VRFs where IPVPN TLVs are used to carry the routes through the network to other SPBM BEB switches.

The following commands are used only when configuring an L3 VPN (ISIS-SPB) configuration (the gateway IP address specified is the IP address of the IPv4 L3 VPN interface).

BEB-A:

```
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.1 all-routes
-> spb ipvpn bind vrf-1 isid 2000 gateway 2.2.2.1 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
vrf-1::-> ip import isid 2000 all-routes
```

BEB-B (OS9900):

```
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.2 all-routes
-> spb ipvpn bind vrf-1 isid 2000 gateway 2.2.2.2 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
vrf-1::-> ip import isid 2000 all-routes
```

IPv4 L3 VPN External Loopback Interfaces: I-SID Routing in One VRF

In this sample IPv4 over SPB configuration, Networks A and B can communicate with each other within the same VRF but on different I-SIDs due to the routing (redistribution) between I-SID 1000 and 2000 on BEB-C. In addition, Network C is also able to communicate with Networks A and B.

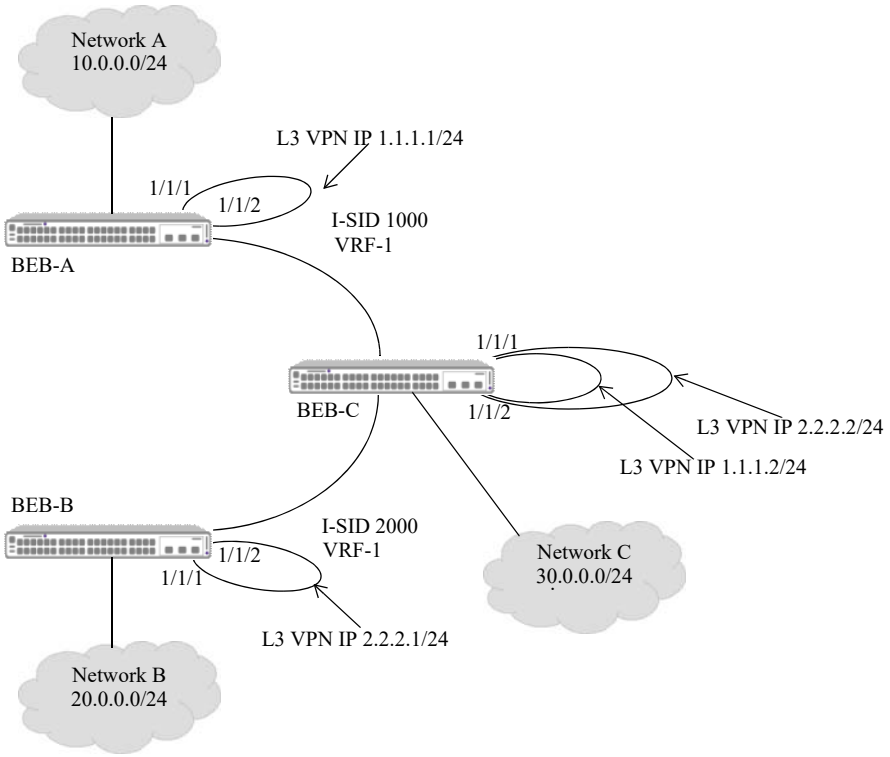


Figure 7-20 : IPv4 Inter-ISID Routing Example (One VRF)

In this topology,

- Network A binds to I-SID 1000 in VRF-1, Network B binds to I-SID 2000 in VRF-1, and Network C binds to both I-SIDs in VRF-1.
- A physical loopback port configuration is required on BEB-A, BEB-B, and BEB-C. Port 1/1/1 is the L3 VPN router port, port 1/1/2 is the L3 VPN access port, and VLAN 200 and VLAN 400 are the L3 VPN VLANs.
  - Port 1/1/1 is tagged with VLAN 200 (BEB-A and BEB-C) and VLAN 400 (BEB-B and BEB-C).
  - Access port 1/1/2 is assigned to SAPs that are each associated with an I-SID (an SPB service instance) that will forward VLAN 200 (BEB-A and BEB-C) and VLAN 400 (BEB-B and BEB-C) traffic through the SPB backbone network.
  - A physical cable is connected to port 1/1/1 and to port 1/1/2 on each switch to create the connection between the VLAN and service domains.
  - On BEB-A, an IPv4 L3 VPN interface is configured on VLAN 200 (1.1.1.1/24).
  - On BEB-B, an IPv4 L3 VPN interface is configured on VLAN 400 (2.2.2.1/24).
  - On BEB-C, an IPv4 L3 VPN interface is configured on VLAN 200 (1.1.1.2/24) and on VLAN 400 (2.2.2.2/24).

The following CLI command examples are used to configure the sample IPv4 over SPB topology shown in "IPv4 Inter-ISID Routing Example (One VRF)" on page 7-91.

BEB-A:

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 vlan 200 address 1.1.1.1/24
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.1 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
```

BEB-B

```
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn2 vlan 400 address 2.2.2.1/24
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 2000 gateway 2.2.2.1 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 2000 all-routes
```

BEB-C:

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 vlan 200 address 1.1.1.2/24
vrf-1::-> ip interface l3vpn2 vlan 400 address 2.2.2.2/24
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.2 all-routes
-> spb ipvpn bind vrf-1 isid 2000 gateway 2.2.2.2 all-routes
-> spb ipvpn redist source-isid 1000 destination-isid 2000 all-routes
-> spb ipvpn redist source-isid 2000 destination-isid 1000 all-routes
-> vrf vrf-1
vrf-1::-> ip import all-routes
vrf-1::-> ip export all-routes
```

IPv6 L3 VPN External Loopback Interfaces: I-SID Routing in One VRF

In this sample IPv6 over SPB configuration, Networks A and B can communicate with each other within the same VRF but on different I-SIDs due to the routing (redistribution) between I-SID 1000 and 2000 on BEB-C. In addition, Network C is also able to communicate with Networks A and B.

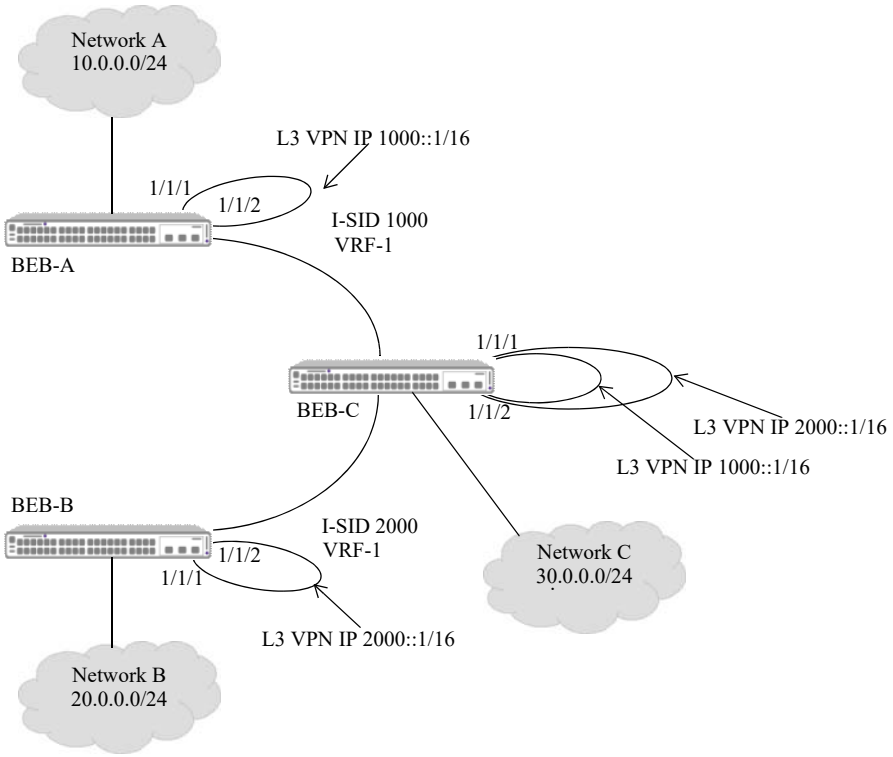


Figure 7-21 : IPv6 Inter-ISID Routing Example (One VRF)

In this topology,

- Network A binds to I-SID 1000 in VRF-1, Network B binds to I-SID 2000 in VRF-1, and Network C binds to both I-SIDs in VRF-1.
- A physical loopback port configuration is required on BEB-A, BEB-B, and BEB-C. Port 1/1/1 is the L3 VPN router port, port 1/1/2 is the L3 VPN access port, and VLAN 200 and VLAN 400 are the L3 VPN VLANs.
  - Port 1/1/1 is tagged with VLAN 200 (BEB-A and BEB-C) and VLAN 400 (BEB-B and BEB-C).
  - Access port 1/1/2 is assigned to SAPs that are each associated with an I-SID (an SPB service instance) that will forward VLAN 200 (BEB-A and BEB-C) and VLAN 400 (BEB-B and BEB-C) traffic through the SPB backbone network.
  - A physical cable is connected to port 1/1/1 and to port 1/1/2 on each switch to create the connection between the VLAN and service domains.
  - On BEB-A, an IPv6 L3 VPN interface is configured on VLAN 200 (1000::1/16).
  - On BEB-B, an IPv6 L3 VPN interface is configured on VLAN 400 (2000::1/16).
  - On BEB-C, an IPv6 L3 VPN interface is configured on VLAN 200 (1000::2/16) and on VLAN 400 (2000::2/16).

The following CLI command examples are used to configure the sample IPv6 over SPB topology shown in "IPv4 Inter-ISID Routing Example (One VRF)" on page 7-91.

BEB-A:

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> vrf create vrf-1
vrf-1::-> ipv6 interface l3vpn1 vlan 200 address 1000::1/16
vrf-1::-> ipv6 export all-routes
vrf-1::-> vrf default
-> spb ipvpn6 bind vrf-1 isid 1000 gateway 1000::1 all-routes
-> vrf vrf-1
vrf-1::-> ipv6 import isid 1000 all-routes
```

BEB-B

```
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-1
vrf-1::-> ipv6 interface l3vpn2 vlan 400 address 2000::1/16
vrf-1::-> ipv6 export all-routes
vrf-1::-> vrf default
-> spb ipvpn6 bind vrf-1 isid 2000 gateway 2000::1 all-routes
-> vrf vrf-1
vrf-1::-> ipv6 import isid 2000 all-routes
```

BEB-C:

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-1
vrf-1::-> ipv6 interface l3vpn1 vlan 200 address 1000::2/16
vrf-1::-> ipv6 interface l3vpn2 vlan 400 address 2000::2/16
vrf-1::-> vrf default
-> spb ipvpn6 bind vrf-1 isid 1000 gateway 1000::2 all-routes
-> spb ipvpn6 bind vrf-1 isid 2000 gateway 2000::2 all-routes
-> spb ipvpn6 redist source-isid 1000 destination-isid 2000 all-routes
-> spb ipvpn6 redist source-isid 2000 destination-isid 1000 all-routes
-> vrf vrf-1
vrf-1::-> ipv6 import all-routes
vrf-1::-> ipv6 export all-routes
```

IPv4 L3 VPN External Loopback Interfaces: I-SID Routing in Two VRFs

In this sample IPv4 over SPB configuration, Networks A and B can communicate with each other between two different VRFs on different I-SIDs due to the routing (redistribution) between I-SID 1000 and 2000 on BEB-C. In addition, Network C is also able to communicate with Networks A and B.

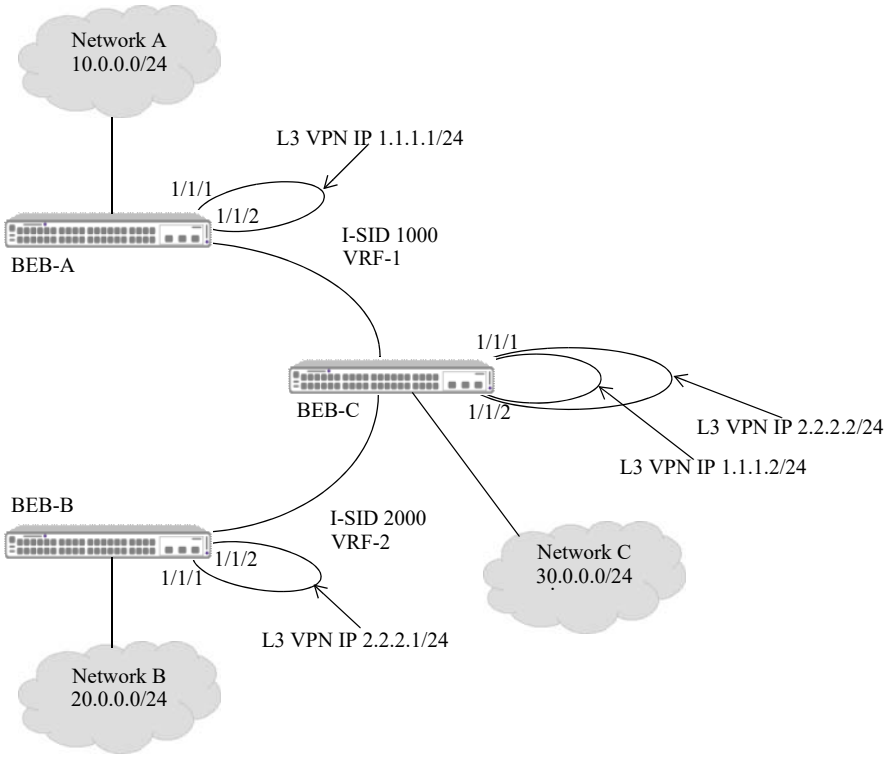


Figure 7-22 : IPv4 Inter-ISID Routing Example (Two VRFs)

In this topology,

- Network A binds to I-SID 1000 in VRF-1, Network B binds to I-SID 2000 in VRF-2, and Network C binds to both I-SIDs in VRF-1.
- A physical loopback port configuration is required on BEB-A, BEB-B, and BEB-C. Port 1/1/1 is the L3 VPN router port, port 1/1/2 is the L3 VPN access port, and VLAN 200 and VLAN 400 are the L3 VPN VLANs.
  - Port 1/1/1 is tagged with VLAN 200 (BEB-A and BEB-C) and VLAN 400 (BEB-B and BEB-C).
  - Access port 1/1/2 is assigned to SAPs that are each associated with an I-SID (an SPB service instance) that will forward VLAN 200 (BEB-A and BEB-C) and VLAN 400 (BEB-B and BEB-C) traffic through the SPB backbone network.
  - A physical cable is connected to port 1/1/1 and to port 1/1/2 to create the connection between the VLAN and service domains.
  - On BEB-A, an IPv4 L3 VPN interface is configured on VLAN 200 (1.1.1.1/24).
  - On BEB-B, an IPv4 L3 VPN interface is configured on VLAN 400 (2.2.2.1/24).
  - On BEB-C, an IPv4 L3 VPN interface is configured on VLAN 200 (1.1.1.2/24) and on VLAN 400 (2.2.2.2/24).

The following CLI command examples are used to configure the sample IPv4 over SPB topology shown in "IPv4 Inter-ISID Routing Example (Two VRFs)" on page 7-95.

BEB-A

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 vlan 200 address 1.1.1.1/24
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.1 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
```

BEB-B

```
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-2
vrf-2::-> ip interface l3vpn2 vlan 400 address 2.2.2.1/24
vrf-2::-> ip export all-routes
vrf-2::-> vrf default
-> spb ipvpn bind vrf-1 isid 2000 gateway 2.2.2.1 all-routes
-> vrf vrf-2
vrf-2::-> ip import isid 2000 all-routes
```

BEB-C

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 vlan 200 address 1.1.1.2/24
vrf-1::-> vrf create vrf-2
vrf-2::-> ip interface l3vpn2 vlan 400 address 2.2.2.2/24
vrf-2::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.2 all-routes
-> spb ipvpn bind vrf-2 isid 2000 gateway 2.2.2.2 all-routes
-> spb ipvpn redist source-isid 1000 destination-isid 2000 all-routes
-> spb ipvpn redist source-isid 2000 destination-isid 1000 all-routes
-> spb ipvpn redist source-vrf vrf-1 destination-isid 2000 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
vrf-1::-> ip import isid 2000 all-routes
vrf-1::-> ip export all-routes
vrf-1::-> vrf vrf-2
vrf-2::-> ip import vrf-1 all-routes
vrf-2::-> ip import isid 1000 all-routes
vrf-2::-> ip import isid 2000 all-routes
```

IPv6 L3 VPN External Loopback Interfaces: I-SID Routing in Two VRFs

In this sample IPv6 over SPB configuration, Networks A and B can communicate with each other between two different VRFs on different I-SIDs due to the routing (redistribution) between I-SID 1000 and 2000 on BEB-C. In addition, Network C is also able to communicate with Networks A and B.

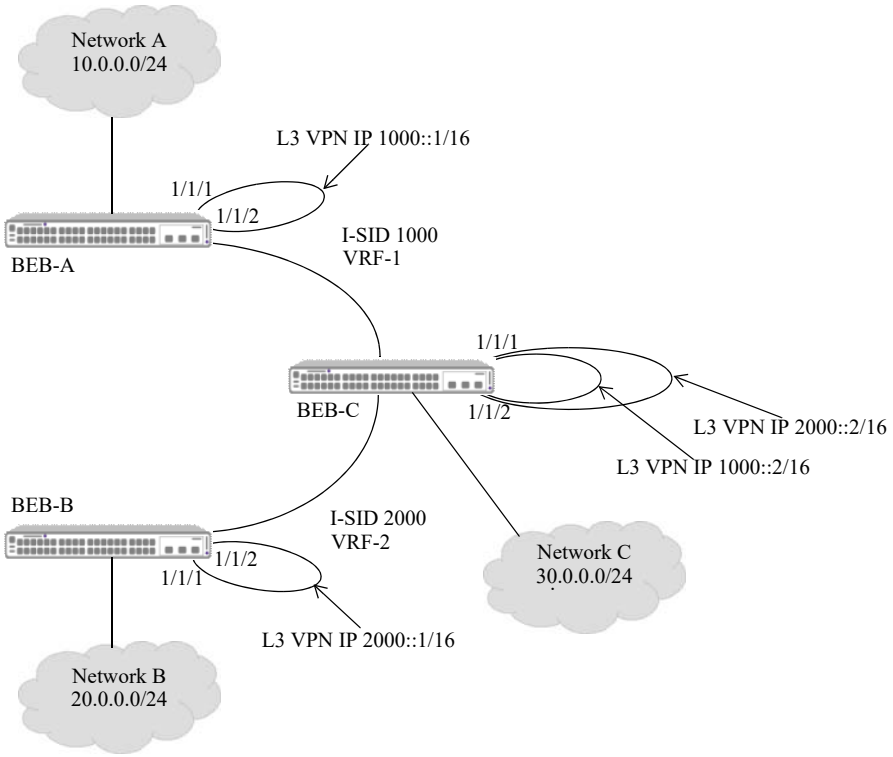


Figure 7-23 : IPv6 Inter-ISID Routing Example (Two VRFs)

In this topology,

- Network A binds to I-SID 1000 in VRF-1, Network B binds to I-SID 2000 in VRF-2, and Network C binds to both I-SIDs in VRF-1.
- A physical loopback port configuration is required on BEB-A, BEB-B, and BEB-C. Port 1/1/1 is the L3 VPN router port, port 1/1/2 is the L3 VPN access port, and VLAN 200 and VLAN 400 are the L3 VPN VLANs.
  - Port 1/1/1 is tagged with VLAN 200 (BEB-A and BEB-C) and VLAN 400 (BEB-B and BEB-C).
  - Access port 1/1/2 is assigned to SAPs that are each associated with an I-SID (an SPB service instance) that will forward VLAN 200 (BEB-A and BEB-C) and VLAN 400 (BEB-B and BEB-C) traffic through the SPB backbone network.
  - A physical cable is connected to port 1/1/1 and to port 1/1/2 to create the connection between the VLAN and service domains.
  - On BEB-A, an IPv6 L3 VPN interface is configured on VLAN 200 (1000::1/16).
  - On BEB-B, an IPv6 L3 VPN interface is configured on VLAN 400 (2000::1/16).
  - On BEB-C, an IPv6 L3 VPN interface is configured on VLAN 200 (1000::2/16) and on VLAN 400 (2000::2/16).

The following CLI command examples are used to configure the sample IPv4 over SPB topology shown in "IPv4 Inter-ISID Routing Example (Two VRFs)" on page 7-95.

BEB-A

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> vrf create vrf-1
vrf-1::-> ipv6 interface l3vpn1 vlan 200 address 1000::1/16
vrf-1::-> ipv6 export all-routes
vrf-1::-> vrf default
-> spb ipvpn6 bind vrf-1 isid 1000 gateway 1000::1 all-routes
-> vrf vrf-1
vrf-1::-> ipv6 import isid 1000 all-routes
```

BEB-B

```
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-2
vrf-2::-> ipv6 interface l3vpn2 vlan 400 address 2000::1/16
vrf-2::-> ipv6 export all-routes
vrf-2::-> vrf default
-> spb ipvpn6 bind vrf-1 isid 2000 gateway 2000::1 all-routes
-> vrf vrf-2
vrf-2::-> ipv6 import isid 2000 all-routes
```

BEB-C

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-1
vrf-1::-> ipv6 interface l3vpn1 vlan 200 address 1000::2/16
vrf-1::-> vrf create vrf-2
vrf-2::-> ipv6 interface l3vpn2 vlan 400 address 2000::2/16
vrf-2::-> vrf default
-> spb ipvpn6 bind vrf-1 isid 1000 gateway 1000::2 all-routes
-> spb ipvpn6 bind vrf-2 isid 2000 gateway 2000::2 all-routes
-> spb ipvpn6 redist source-isid 1000 destination-isid 2000 all-routes
-> spb ipvpn6 redist source-isid 2000 destination-isid 1000 all-routes
-> spb ipvpn6 redist source-vrf vrf-1 destination-isid 2000 all-routes
-> vrf vrf-1
vrf-1::-> ipv6 import isid 1000 all-routes
vrf-1::-> ipv6 import isid 2000 all-routes
vrf-1::-> ipv6 export all-routes
vrf-1::-> vrf vrf-2
vrf-2::-> ipv6 import vrf-1 all-routes
vrf-2::-> ipv6 import isid 1000 all-routes
vrf-2::-> ipv6 import isid 2000 all-routes
```

L3 VPN External Loopback Migration to Inline Routing

In the following sample IP over SPB topology, an external loopback configuration serves as an L3 VPN gateway interface.

- An IP interface ("L3VPN-10") is created in a specific VRF instance and bound to VLAN 400.
- Bridge port 1/1/6 is tagged with VLAN 400.
- Service access port 1/1/1 is assigned to an SPB SAP that will process customer traffic that is tagged with VLAN 400 (1/1/1:400). The SAP is bound to SPB service 10.
- Service access port 1/1/5 is assigned to an SPB SAP that will process traffic tagged with VLAN 400 (SAP 1/1/5:400). The SAP is bound to SPB service 10.
- A physical cable connects VLAN port 1/1/6 with service access port 1/1/5 to form the loopback configuration. Traffic is passed through the physical loopback cable between the Layer 2 SPB service domain and the Layer 3 VLAN routing domain.
  - The service access port handles Layer 2 bridging into the service domain (between SAPs 1/1/1:400 and 1/1/5:400).
  - The VLAN port handles the Layer 3 routing into the VLAN domain (between VLAN 400 and other IP interface VLANs).

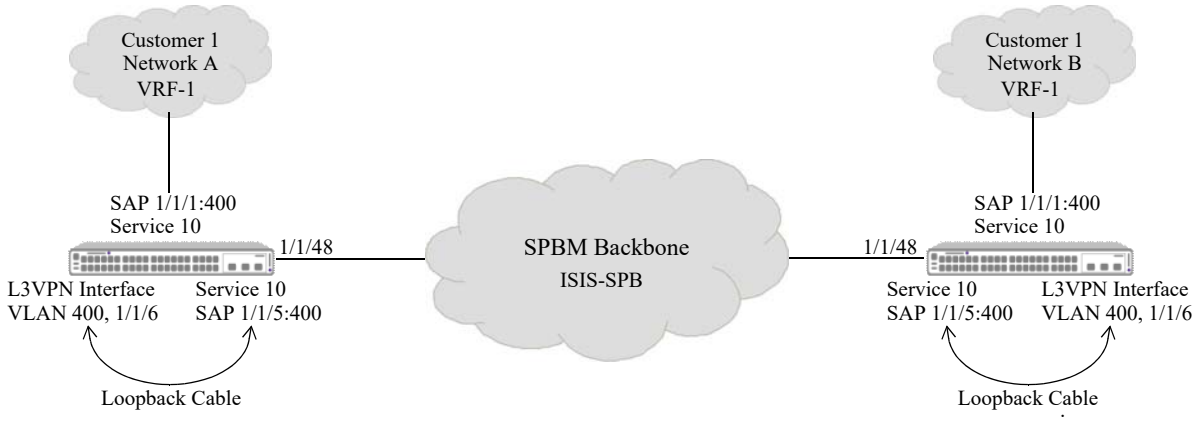


Figure 7-24 : L3 VPN Interface Before Migration to Inline Routing

The following CLI command examples are used to configure the sample external loopback configuration:

```
-> spb interface port 1/1/48
-> spb bvlan 4000
-> spb isis control-bvlan 4000
-> service access port 1/1/1 vlan-xlation enable
-> service spb 10 isid 1000 bvlan 4000 vlan-xlation enable
-> service 10 sap port 1/1/1:400
-> service access port 1/1/5 vlan-xlation enable
-> service 10 sap port 1/1/5:400
-> vlan 400
-> vlan 400 member port 1/1/6 tagged
-> ip interface L3VPN-10 address 10.0.0.1/24 vlan 400
```

To change the L3 VPN interface setup to use service-based inline routing instead of an external loopback, the following steps are required:

1. Delete the IP interface assigned to VLAN 400.

```
-> no ip interface L3VPN-10
```

2. Delete VLAN 400 (this automatically removes the VLAN 400 tag from port 1/1/6).

```
-> no vlan 400
```

3. Delete SAP 1/1/5:400 (this does not remove SPB service 10).

```
-> no service 10 sap port 1/1/5:400
```

4. Disconnect the physical cable from port 1/1/5 and port 1/1/6.

5. Configure an IP interface and bind it to SPB service 10.

```
-> ip interface L3vpn-10 address 10.0.0.1/24 service 10
```

The following shows the sample IP over SPB topology after the L3 VPN interface setup has been changed from an external loopback configuration to a service-based inline routing configuration.

Figure 7-25 : L3 VPN Interface After Migration to Inline Routing

Refer to "IP over SPB Configuration Guidelines" on page 7-73 for more information about configuring an L3 VPN gateway interface.

Configuring SPB Over Shared Ethernet

Configuring SPB multi-access network interfaces to implement the SPB over shared Ethernet feature was not supported prior to AOS release 8.7R1; only point-to-point (P2P) interfaces were supported. In a network where some SPB nodes are running AOS release 8.7R1 and other SPB nodes are running earlier AOS releases, configuring multi-access interfaces may cause inconsistent connectivity to destinations beyond the shared Ethernet network segment. If such network reachability is desired, those SPB nodes must be upgraded to AOS Release 8.7R1.

IS-IS is a link state protocol that fundamentally requires every participating device in the topology to have the same view of the topology represented by a link state database and to follow the same procedure for determining the shortest path to reach every other participating device in the network.

The SPB over shared Ethernet feature represents every shared link/network as a pseudo-node. One of the nodes on the shared link is elected as a Designated IS (DIS) node which originates a pseudo-node LSP. This pseudo-node LSP needs to be flooded across the entire SPB network and all the devices in the SPB network (not just the devices directly connected to the shared network) need to understand this LSP to determine the shortest paths to other devices which travels through the shared network.

Advertisement and processing of such pseudo-node LSPs is an enhancement to the standardized ISIS-SPB functionality provided through the ability to configure SPB multi-access links. AOS releases prior to 8.7R1 only supported P2P links and do not process pseudo-node LSPs.

SPB Over Shared Ethernet Configuration Guidelines

Consider the following guidelines when configuring an SPB backbone over a shared Ethernet domain, as described in "SPB Over Shared Ethernet" on page 7-23:

- Identify the Backbone Edge Bridges (BEBs) that are connected to the multi-access domain (shared network) and on which adjacencies will be formed over the shared network. On each of these BEBs, configure an SPB multi-access network interface.
- Configuring more than one multi-access network interface on the same switch is supported. However, ISIS-SPB selects only one interface on which to form adjacencies.
- Configuring a P2P network interface and a multi-access network interface on the same switch is supported.
- A Designated Intermediate System (DIS) election process determines which multi-access interface will serve as the DIS for the virtual SPB node. A DIS is elected based on which interface has the highest priority value, with the highest backbone MAC address (BMAC) used as a tiebreaker.
- The configurable priority value applies only to SPB multi-access interfaces, not SPB P2P interfaces.
- The DIS defines and represents all the multi-access links as a virtual SPB node (pseudo-node); to the IS-IS network, these links are treated as just another SPB node.
- The SPB configuration requirements for a BEB with P2P interfaces are the same for a BEB with multi-access interfaces. The SPB service objects (BVLANs, service IDs, etc.) configured on each BEB must be the same. ISIS-SPB forms adjacencies based on the underlying SPB backbone configuration for both P2P and multi-access domains.

Configuring an SPB Multi-Access Interface

In order for BEBs to communicate with other BEBs over a multi-access domain, each BEB needs to form adjacencies with all of the other BEBs. This is not possible with a P2P configuration, so each SPB network interface port must be configured as a multi-access interface to allow multiple adjacencies to form across the broadcast network domain.

By default, the SPB network interface type is set to P2P. To configure a multi-access network interface, use the **spb isis interface** command with the **multi-access** option. For example:

```
-> spb isis interface port 1/1/20 type multi-access
```

To configure a P2P network interface, use the **spb isis interface** command with the **p2p** option. For example:

```
-> spb isis interface port 1/1/20 type p2p
```

By default, the priority value for a multi-access interface is set to 64. This value is used to determine which multi-access interface is elected as the DIS. To change this value, use the **spb isis interface** command with the **priority** parameter. For example:

```
-> spb isis interface port 1/1/20 type multi-access priority 100
```

Verifying the SPB Multi-Access Interface Configuration

Each BEB that will form adjacencies over a shared Ethernet domain must be configured with an SPB multi-access interface. To verify the interface type, use the **show spb isis interface** command. The “Circ Type” field displays the interface type (“P2P or “Multi-Access”). For example:

```
-> show spb isis interface
SPB ISIS Interfaces:
                                Oper   Admin   Link      Hello   Hello   Circ
 Interface      Level  CircID  state  state   Metric    Intvl   Mult    Type
--------------+-------+-------+------+-------+-------+-------+-------+------------
 1/1/1          L1      1       DOWN   UP      10      9       3       P2P
 1/1/2          L1      2       UP     UP      10      9       3       P2P
 1/1/3          L1      3       DOWN   UP      10      9       3       P2P
 1/1/4          L1      4       DOWN   UP      10      9       3       P2P
 1/1/5          L1      5       UP     UP      10      3       3       Multi-Access
 1/1/6          L1      6       DOWN   UP      10      9       3       P2P
 1/1/7          L1      7       DOWN   UP      10      9       3       P2P
 1/1/10         L1      9       DOWN   UP      10      9       3       P2P
 0/3            L1      1       UP     UP      10      9       3       P2P

Interfaces : 9
```

To determine if the multi-access interface is serving as the DIS, use the **show spb isis interface** command with the **port** or **linkagg** parameter. For example, the following displays additional information for the SPB interface configured on port 1/1/5:

```
-> show spb isis interface port 1/1/5
-------------------------------------------------------------------------------
Interface        : 1/1/5                     Type          : Multi-Access
Oper State       : UP                        Admin State   : UP
Circuit Id       : 1                         CSNP Int      : 10 sec
Desg IS          : 00d0.9501.8a1c            Adjacencies   : 3
Metric           : 10                        Hello Timer   : 3 sec
Hello Mult       : 3                         Priority      : 64
-------------------------------------------------------------------------------
```

In the above **show** command example, interface port 1/1/5 is serving as the DIS. This is indicated by the system ID displayed in the “Desg IS” field. If this interface is not a multi-access interface, then “N/A” would appear in this field.

To verify the adjacencies formed on a multi-access interface, use the **show spb isis adjacency** command. For example, the following displays three adjacencies discovered and formed on multi-access interface port 1/1/5:

```
-> show spb isis adjacency
SPB ISIS Adjacency:
System
 (Name : SystemId)                      Type   State   Hold  Interface
-------------------------------------+------+-------+------+----------
BEB-3                : 00d0.9501.f02c  L1    UP        25      1/1/5
BEB-4                : 00d0.9501.f61c  L1    UP        18      1/1/5
BEB-5                : 00d0.9501.51dc  L1    UP        26      1/1/5
```

The elected DIS represents all of the multi-access links as a virtual SPB node (pseudo-node). As a result, the DIS also generates a pseudo-node LSP that lists all the multi-access links as neighbors. The pseudo-node LSP differs from a regular LSP in that a regular LSP lists the pseudo-node as a neighbor, along with other information about the SPB service configuration.

To display the ISIS-SPB topology information maintained in the link state database (LSDB), use the **show spb isis database** command. For example:

```
-> show spb isis database
Legends : P    = The Partition repair bit is set
          OV   = The overload bit is set
          ATT  = The Attach bit is set
          L1   = Specifies a Level 1 IS type
          L2   = Specifies a Level 2 IS type
SPB ISIS LSP Database:
LSP ID                 Sequence    Checksum   Lifetime   Attributes
----------------------+-----------+----------+----------+------------
 00d0.9501.51dc.00-00        0x0b     0x98ba       1049   L1
 00d0.9501.8a1c.00-00        0x0d     0x9705       1047   L1
 00d0.9501.8a1c.23-00        0x01     0x7e66        985   L1
 00d0.9501.f02c.00-00        0x07     0x515b        990   L1
 00d0.9501.f02c.19-00        0x01       0x00      (982)   L1
 00d0.9501.f61c.00-00        0x06     0xcc08        991   L1
 00d0.9501.f61c.15-00        0x03       0x00      (984)   L1

 Level-1 LSP count : 7
```

To display the pseudo-node LSP, use the **show spb isis database** command with the **lsp-id** parameter. For example, the following command displays pseudo-node LSP ID 00d0.9501.8alc.23-00, which was generated by the DIS multi-access interface:

```
-> show spb isis database lsp-id 00d0.9501.8a1c.23-00
Legends : P    = The Partition repair bit is set
          OV   = The overload bit is set
          ATT  = The Attach bit is set
          L1   = Specifies a Level 1 IS type
          L2   = Specifies a Level 2 IS type
SPB ISIS LSP Database:
-------------------------------------------------------------------------------
LSP ID        : 00d0.9501.8a1c.23-00                    Level     : L1
Sequence      : 0x01             Checksum  : 0x7e66     Lifetime  : 966
Version       : 1                Pkt Type  : 18         Pkt Ver   : 1
Attributes    : L1               Max Area  : 3
```

```
SysID Len     : 6                Used Len  : 105       Alloc Len : 1492

TLVs :
TE IS Neighbors     :
   Neighbor         : 00d0.9501.8a1c  SPB Metric 10 Num of Ports 1 Port-Id 0x8002()
   Neighbor         : 00d0.9501.51dc  SPB Metric 10 Num of Ports 1 Port-Id 0x8002()
   Neighbor         : 00d0.9501.f02c  SPB Metric 10 Num of Ports 1 Port-Id 0x8002()
   Neighbor         : 00d0.9501.f61c  SPB Metric 10 Num of Ports 1 Port-Id 0x8002()
```

SPB Over Shared Ethernet Configuration Examples

An SPB multi-access domain can be configured over a shared Ethernet network (such as a service provider network) or even another SPB network. This section contains diagrams and CLI command examples for configuring the following scenarios:

- “SPB Backbone over a Shared Network” on page 7-105.
- “SPB Backbone over Another SPB Network” on page 7-106.

SPB Backbone over a Shared Network

The following diagram shows an example of an SPB network backbone comprised of SPB Backbone Edge Bridges (BEBs) that are connected to a shared service provider network. In this sample topology, each BEB establishes ISIS-SPB adjacencies with all of the other BEBs to form an SPB backbone that is extended over the service provider network.

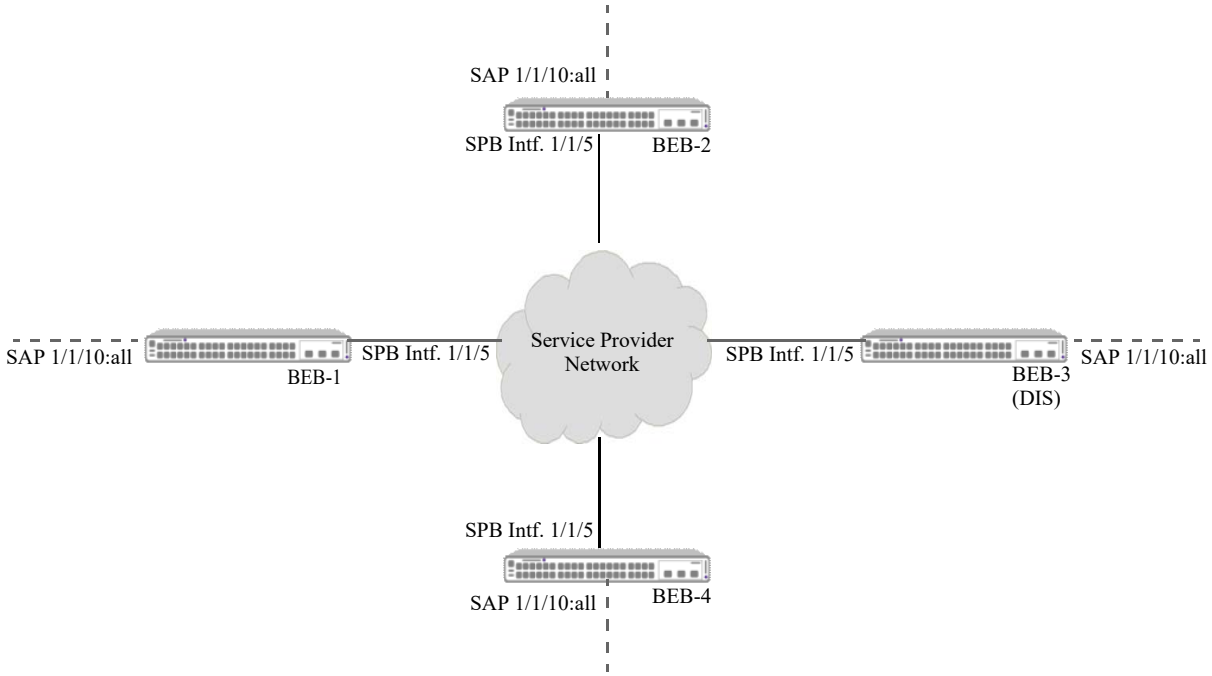


Figure 7-26 : SPB Backbone over a Shared Network

In this sample topology, all four BEBs are configured as follows:

- BVLAN 4001 and 4002, with Control BVLAN set to 4001.
- SPB network interface port 1/1/5 configured as a multi-access interface.
- Port 1/1/10 configured as an access port
- SPB service 20 bound to I-SID 1501 and BVLAN 4001.
- SPB service 40 bound to I-SID 1502 and BVLAN 4002.
- A service access point (SAP) comprised of the SPB service, access port, and encapsulation. The SAP identifies VLAN traffic to associate with the service.

The following CLI command examples are used to configure each BEB in the sample topology:

```
-> spb bvlan 4001
-> spb bvlan 4002
-> spb isis control-bvlan 4001
-> spb isis interface port 1/1/5 type multi-access
-> service access port 1/1/10
-> service 20 spb isid 1501 bvlan 4001
-> service 40 spb isid 1502 bvlan 4002
-> service 20 sap port 1/1/10:all
-> service 40 sap port 1/1/10:all
-> spb isis admin-state enable
```

SPB Backbone over Another SPB Network

The following diagram shows an example of an SPB network backbone comprised of SPB Backbone Edge Bridges (BEBs) that are connected to another SPB network. In this sample topology, the SPB network interface port on each BEB connects to an access port on the inner SPB backbone. ISIS-SPB adjacencies are established between all of the BEBs over the inner SPB network.

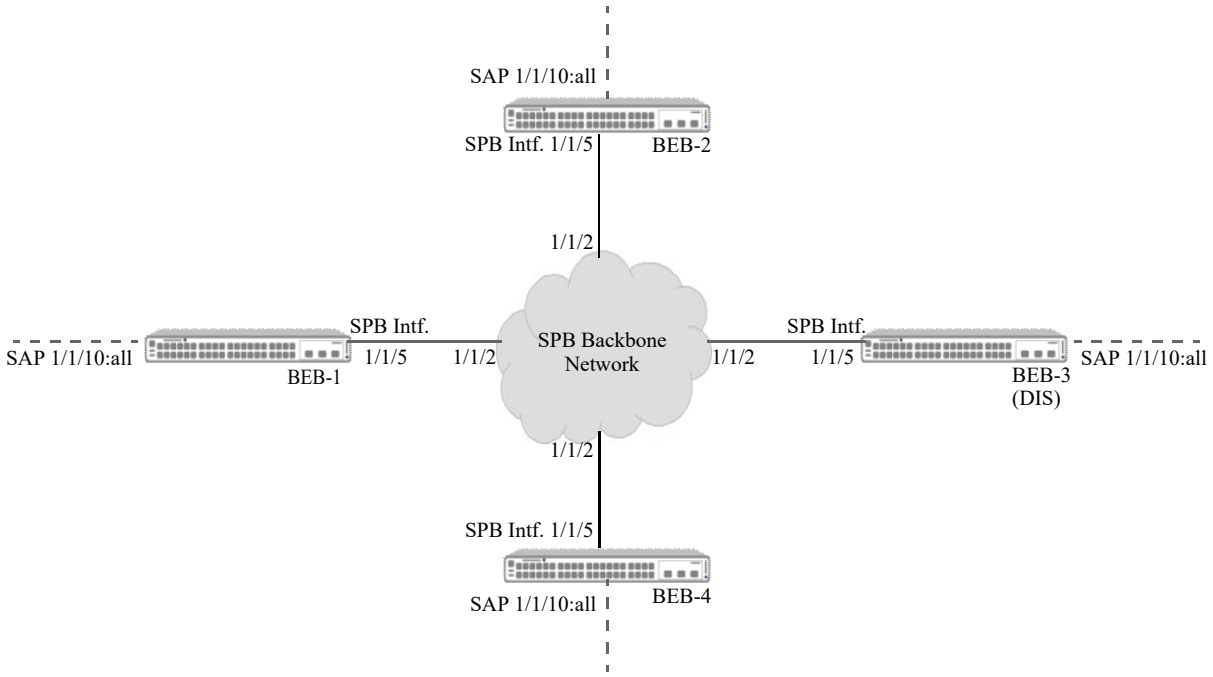


Figure 7-27 : SPB Backbone over Another SPB Network

In this sample topology, all four BEBs are configured as follows:

- BVLAN 4001 and 4002, with Control BVLAN set to 4001.
- SPB network interface port 1/1/5 configured as a multi-access interface. Each multi-access interface port connects to access port 1/1/2 on the inner SPB backbone network.
- Port 1/1/10 configured as an access port.
- SPB service 20 bound to I-SID 1501 and BVLAN 4001.
- SPB service 40 bound to I-SID 1502 and BVLAN 4002.
- A Service Access Point (SAP) comprised of the SPB service, access port, and encapsulation. The SAP identifies VLAN traffic to associate with the service.

The following CLI command examples are used to configure each BEB in the sample topology:

```
-> spb bvlan 4001
-> spb bvlan 4002
-> spb isis control-bvlan 4001
-> spb isis interface port 1/1/5 type multi-access
-> service access port 1/1/10
-> service 20 spb isid 1501 bvlan 4001
-> service 40 spb isid 1502 bvlan 4002
-> service 20 sap port 1/1/10:all
-> service 40 sap port 1/1/10:all
-> spb isis admin-state enable
```

Verifying the SPB Backbone and Services

Displaying the SPBM configuration is helpful to verify the actual configuration on each SPB switch in the topology and to troubleshoot ISIS-SPB backbone and SPB service connectivity.

Verifying the ISIS-SPB Backbone Configuration

To display information about the ISIS-SPB infrastructure (backbone), use the **show** commands listed in this section.

show spb isis info	Displays the global status and configuration for the ISIS-SPB instance on the switch.
show spb isis bvlans	Displays the backbone VLAN (BVLAN) configuration for the switch.
show spb isis interface	Displays the SPB interface (network port) configuration for the switch.
show spb isis adjacency	Displays information about the ISIS-SPB adjacencies created for the SPB switch.
show spb isis database	Displays ISIS-SPB topology information maintained in the link state database (LSDB).
show spb isis nodes	Displays the discovered node-level parameter values for all of the ISIS-SPB switches participating in the topology.
show spb isis unicast-table	Displays the unicast forwarding information for the BVLAN topology.
show spb isis services	Displays a network-wide view of existing services to help verify that SPB services are correctly advertised and learned by ISIS-SPB
show spb isis spf	Displays the shortest path first (SPF) information to all known SPB switches for a specific BVLAN.
show spb isis multicast-table	Displays the multicast forwarding entries for services.
show spb isis multicast-sources	Displays all the known multicast sources across the SPB domain and BVLANs.
show spb isis multicast-sources-spf	Displays the shortest path first (SPF) readability for a known multicast source bridge for a specific BVLAN.
show spb isis ingress-mac-filter	Displays the ingress MAC filter for multicast traffic for a given BVLAN operating in the (*,G) mode.

Verifying the SPB Service Configuration

To display information about the Service Manager configuration for SPB service connectivity, use the **show** commands listed in this section.

show service access	Displays the service access (customer-facing) port configuration.
show service l2profile	Displays the Layer 2 profile definitions. These profiles are applied to service access ports to determine how Layer 2 control protocol frames are processed on these ports.
show service	Displays the service configuration.
show service ports	Displays all the virtual ports (SAPs, SDPs) that are associated with an SPB service.
show service spb sap	Displays the configuration information for the specified SAP ID associated with the specified service.
show service sdp	Displays the dynamic Service Distribution Point (SDP) configuration.
show service bind-sdp	Displays the dynamic SDP-to-service binding configuration.
show service debug-info	Displays debug information for the virtual ports associated with the SPB service.
show service info	Displays the global Service Manager configuration for the switch.
show service counters	Displays the traffic statistics for the specified SPB service and associated virtual ports.
clear service counters	Clears the traffic statistics for the specified SPB service and associated virtual ports.

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

8 Configuring Loopback Detection

Loopback Detection (LBD) automatically detects the loop and shutdown the port involved in the loop. This prevents forwarding loops on ports that have forwarded network traffic which has looped back to the originating switch. LBD detects and prevents Layer 2 forwarding loops on a port either in the absence of other loop detection mechanisms such as STP/RSTP/MSTP, or when these mechanisms cannot detect it (for example, a client's equipment may drop BPDUs, or the STP protocol may be restricted to the network edge).

A provider network with a set of multiple switches interconnected together can be logically viewed as a large single switch. The large single switch provides service access points to customers' networks. Configuration faults in customer networks can result in loops spanning both provider and customer networks. This can result in broadcast storms. In order to protect provider's network from broadcast storms, loops that involve SAP ports need to be detected and broken.

The LBD can detect and break loops created on the service-access interface.

For a service-access interface, LBD can be enabled for a specific port or linkagg. LBD for service-access points allows shutting down only the specific interface of the link involved in the loop.

The switch can be configured to process LBD frames received from a different or remote system. The port of the remote system is shut down, rather than passing it as invalid LBD frames.

When loopback occurs, a trap is sent and the event is logged. The port which is shutdown due to LBD is automatically recovered if autorecovery-timer is set or the port can manually be enabled again when the problem is resolved.

In This Chapter

This chapter describes the LBD feature and how to configure it through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. This chapter provides an overview of LBD and includes the following information:

- “Quick Steps for Configuring LBD” on page 8-3
- “LBD Overview” on page 8-4
- “Configuring LBD” on page 8-7
- “LBD for Service Access Interface” on page 8-8
- “Verifying the LBD Configuration” on page 8-12

LBD Defaults

The following table shows LBD default values.

Parameter Description	Command	Default Value/Comments
LBD administrative state	**loopback-detection**	Disabled
LBD remote-origin administrative state	**loopback-detection**	Disabled
LBD status of a port	**loopback-detection port**	Disabled
Remote-origin LBD status of a port	**loopback-detection port**	Disabled
LBD service-access state	**loopback-detection service-access**	Disabled
Transmission time is the time period between LBD packet transmissions.	**loopback-detection service-access**	30 seconds
Autorecovery time is the time period in which the switch is recovered from the shutdown state.	**loopback-detection autorecovery-timer**	300 seconds

Quick Steps for Configuring LBD

The following steps provide a quick tutorial on how to configure LBD. Each step describes a specific operation and provides the CLI command syntax for performing that operation.

1 To enable the LBD protocol on a switch, use the **loopback-detection** command. For example:

```
-> loopback-detection enable
```

2 To enable the LBD protocol on a port, use the **loopback-detection port** command. For example:

```
-> loopback-detection port 1/1/2 enable
```

Note. Once the default LBD is enabled on the switch, the remote-origin LBD can be configured. It can be enabled globally or on a per port using the **loopback-detection** command with the remote-origin parameter. For example:

```
-> loopback-detection remote-origin enable
-> loopback-detection port 1/1/2 remote-origin enable
```

---

3 Configure the LBD transmission timer by using the **loopback-detection transmission-timer** command. For example:

```
-> loopback-detection transmission-timer 200
```

4 To change the auto-recovery timer for Loopback detection, use the command **loopback-detection autorecovery-timer.** By default, the violation recovery time is 300 seconds.

```
-> loopback-detection autorecovery-timer 600
```

Note. *Optional*. To verify the LBD global configuration use the **show loopback-detection** command or to verify the LBD configuration on a port use the **show loopback-detection port** command. For example:

```
-> show loopback-detection
Global LBD Status                : enabled,
Global Remote-origin LBD Status  : enabled,
Global LBD Transmission Timer    : 200 sec,
Global LBD Auto-recovery Timer   : 600 sec,

-> show loopback-detection port 1/1/2
Global LBD Status                  : enabled,
Global Remote-origin LBD Status    : enabled,
Global LBD Transmission Timer      : 200 sec,
Global LBD Auto-recovery Timer     : 600 sec,
Port LBD Status                    : enabled,
Port Remote-origin LBD Status      : enabled,
Port LBD State                     : Inactive,
Port LBD Type                      : normal-edge,
```

To verify the LBD statistics of a port, use the **show loopback-detection statistics port** command. For example:

```
-> show loopback-detection statistics port 1/1/2
LBD Port Statistics
LBD Packet Send               : 1
Invalid LBD Packet Received   : 0
Member of Link Aggregation    : -
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in this display.

LBD Overview

Loopback Detection (LBD) automatically detects and prevents L2 forwarding loops on a port. LBD operates in addition to STP which detects forwarding loops. When a loopback is detected, the port is disabled and goes into a shutdown state. A trap is sent and the event is logged.

When enabling and configuring Loopback Detection:

- Enable Loopback Detection globally on the switch.
- Enable Loopback Detection on edge port.

The switch periodically sends out LBD frame from loopback detection enabled port and concludes that the port is looped back if it receives the frame on any of the loop-back detection enabled ports.

Remote-origin LBD can be enabled and configured per port to process the LBD frames received from a remote system.

For service-access ports, LBD detects the loop for all the LBD edge ports involved.

Transmission Timer

Transmission timer is the time duration in seconds at which the port sends LBD frame on the link. When any port is getting blocked due to loopback detection, there will be no further transmission and receiving of any traffic on the blocked port. The port will be go to shutdown state.

By default, the transmission timer for loopback detection is 30 seconds.

Remote-origin LBD Overview

The remote-origin LBD processes the LBD frames originating from a remote system. The frame is processed and the receiving port is moved to shut down state.

The remote-origin LBD is functional, only if both default LBD and remote-origin LBD are enabled globally and at interface level. For the remote-origin LBD to operate:

- Default LBD must be enabled globally
- Remote-origin LBD must be enabled globally
- Default LBD must be configured on the interface
- Remote-origin LBD must be configured on the LBD enabled interface

The following scenario shows the operation of the remote-origin LBD functionality:

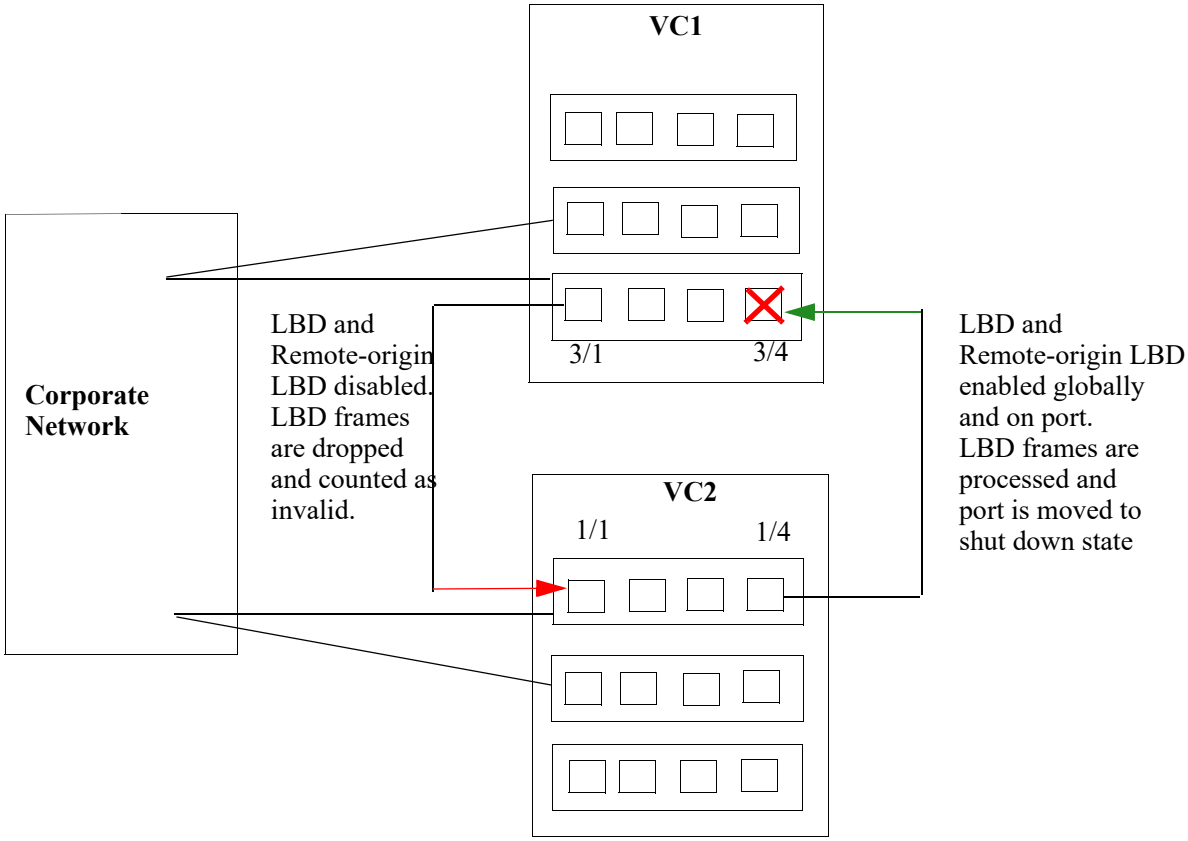


Figure 8-1 : Remote-origin LBD Overview

In the two systems VC1 and VC2, VC1 has both default LBD and remote origin LBD enabled globally and at the interface level (3/4). On VC2 only the default LBD is enabled globally and at interface level.

When LBD frame is transmitted from VC2 (1/4) to VC1 (3/4) the remote LBD frame is processed in VC1, the MAC address of the transmitting system is recorded and the receiving port (3/4) is moved to shut down.

When LBD frame is transmitted from VC1 (3/1) to VC2 (1/1) the LBD frame is dropped as the remote-origin LBD is not enabled.

Note. In case, if remote-origin LBD is enabled on both the systems, the system which receives the first remote LBD frame will shut down the port.

Interaction With Other Features

This section contains important information about how other OmniSwitch features interact with LBD. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

Spanning Tree Protocol

- If the STP mode is set to Multiple Spanning Tree, Loopback Detection can only be enabled on interfaces where STP is disabled.
- LBD frame are sent untagged regardless of the spanning tree state on the port.

Link Aggregation

When loopback is detected on any one of the Linkagg port, all the ports of the linkagg will be shutdown due to loopback detection.

Configuring LBD

This section describes how to use the OmniSwitch Command Line Interface (CLI) commands to configure LBD on a switch.

- Enable LBD on a switch or port (see "Enabling LBD" on page 8-7)
- Enable remote-origin LBD on a switch or port (see"Enabling Remote-origin LBD" on page 8-7)
- Configure the LBD transmission timer (see "Configuring the LBD Transmission Timer" on page 8-8)
- View the LBD statistics on a port (see "Viewing LBD Statistics" on page 8-8)
- Recover a port from LBD shutdown (see "Recovering a Port from LBD Shutdown" on page 8-8)
- Configuring Autorecovery-timer (see "Configuring Autorecovery-timer for LBD Shutdown Ports" on page 8-8)
- Enable LBD on Service Access Interface (see "Enabling LBD on Service-access Interface" on page 8-9)

Enabling LBD

By default, LBD is disabled on the switch. To enable LBD on a switch, use the **loopback-detection** command. For example, the following command enables LBD on a switch:

```
-> loopback-detection enable
```

Enabling LBD on a Port

By default, LBD is disabled on all switch ports. To enable LBD on a port, use the **loopback-detection port** command. For example, the following command enables LBD in chassis 1 on port 1 of slot 1:

```
-> loopback-detection port 1/1/1 enable
```

To enable LBD on multiple ports, specify a range of ports. For example:

```
-> loopback-detection port 1/1/1-8 enable
```

Enabling Remote-origin LBD

By default, remote-origin LBD is disabled on the switch. To enable remote-origin LBD on a switch, use the **loopback-detection** command with the **remote-origin** parameter. For example, the following command enables remote-origin LBD on a switch:

```
-> loopback-detection remote-origin enable
```

Enabling Remote-origin LBD on a port

By default, remote-origin LBD is disabled on all switch ports. To enable remote-origin LBD on a port, use the **loopback-detection port** command with the **remote-origin** parameter. For example, the following command enables remote-origin LBD in chassis 3 on port 1 of slot 1:

```
-> loopback-detection port 3/1/1 remote-origin enable
```

To enable remote-origin LBD on multiple ports, specify a range of ports. For example:

```
-> loopback-detection port 3/1/1-8 remote-origin enable
```

Note. See "Remote-origin LBD Overview" on page 8-4 for more details.

Configuring the LBD Transmission Timer

To configure the transmission time period between LBD packet transmissions, use the **loopback-detection service-access** command. For example:

```
-> loopback-detection transmission-timer 200
```

Viewing LBD Statistics

To view the LBD statistics on a specific port, use the **show loopback-detection statistics port** command. For example, to view the statistics for port 1 on slot 1 of chassis 1, enter:

```
-> show loopback-detection statistics port 1/1/1
```

Recovering a Port from LBD Shutdown

To bring a port out of the shutdown state, use the **interfaces fec** command. For example, to bring the chassis 1, port 5 on slot 1 out of the shutdown state, enter:

```
-> clear-violation port 1/1/5
```

To bring multiple ports out of the shutdown state, enter:

```
-> clear-violation port 1/5/5-10
```

Configuring Autorecovery-timer for LBD Shutdown Ports

The port which is shutdown due to LBD can be automatically recovered if autorecovery-timer is set for the switch. To set the autorecovery-timer, use the **loopback-detection autorecovery-timer** command. For example, to set a autorecovery-timer of 200 sec for the switch, enter:

```
-> loopback-detection autorecovery-timer 200
```

LBD for Service Access Interface

A provider network with a set of multiple switches interconnected together can be logically viewed as a large single switch. The large single switch provides service access points to customers' networks. Configuration faults in customer networks can result in loops spanning both provider and customer networks. This can result in broadcast storms. In order to protect provider's network from broadcast storms, loops that involve SAP ports need to be detected and broken.

The LBD can detect and break loops created on the service-access interface.

For a service-access interface, LBD can be enabled for a specific port or linkagg. LBD for service-access points allows shutting down only the specific interface of the link involved in the loop.

Enabling LBD on Service-access Interface

By default, LBD is disabled for the switch and on all service-access ports. To globally enable LBD for the switch, use the **loopback-detection** command. For example:

```
-> loopback-detection enable
```

To enable LBD on a service-access port, use the **loopback-detection service-access** command. For example:

```
-> loopback-detection service-access port 1/1/1 enable
```

LBD can also be enabled on link aggregates that are configured as service-access aggregates. For example, the following command enables LBD on link aggregate 1:

```
-> loopback-detection service-access linkagg 1 enable
```

Consider the following when configuring LBD on a link aggregate:

- The link aggregate must be formed by ports with same path cost.
- LBD cannot be configured on linkagg which has member ports running LBD configuration and vice versa.
- When a linkagg is in violation or shutdown state, the member ports cannot be deleted from the linkagg.

Note. Before configuring the LBD using the “service-access” option, the port or link aggregate must be configured for service access. Use the **service access** command, to configure the port or link aggregate for service access. When LBD is enabled on ports without the '**service-access**' keyword, the LBD behaves as normal LBD feature.

LBD Packet Processing Mechanism for LBD Service Access Ports

The LBD packets on the service access ports are processed as follows:

1 If Virtual Private (VP) or Virtual Forwarding Instance (VFI) information is present in the packet driver, then the LBD packet is processed else the packet is dropped.

2 The initiator session identifier (ISID) of the packet is extracted from the VP or VFI information and compared with the LBD packet TLV ISID. If the ISID does not match, the packet is dropped.

3 If ISID matches, then the LBD packet TLV path cost is compared with the receiving LBD service access port path cost. If the LBD path cost is lesser, the receiving access port is shut down. If LBD path cost is higher, then the packet is dropped.

4 If path costs are equal, then LBD packet bridge MAC and receiving access port bridge MAC is compared. If LBD packet bridge MAC is lesser, then the receiving access port is shutdown. If LBD Bridge MAC is higher, then the packet is dropped.

5 If LBD packet bridge MAC and receiving access port bridge MAC are same (that is, same switch), then LBD packet port ID and access port ID is compared. If LBD packet port ID is lesser, then the receiving access port is shutdown. If LBD packet port ID is higher, then the packet is dropped.

Sample Scenarios

Scenario 1

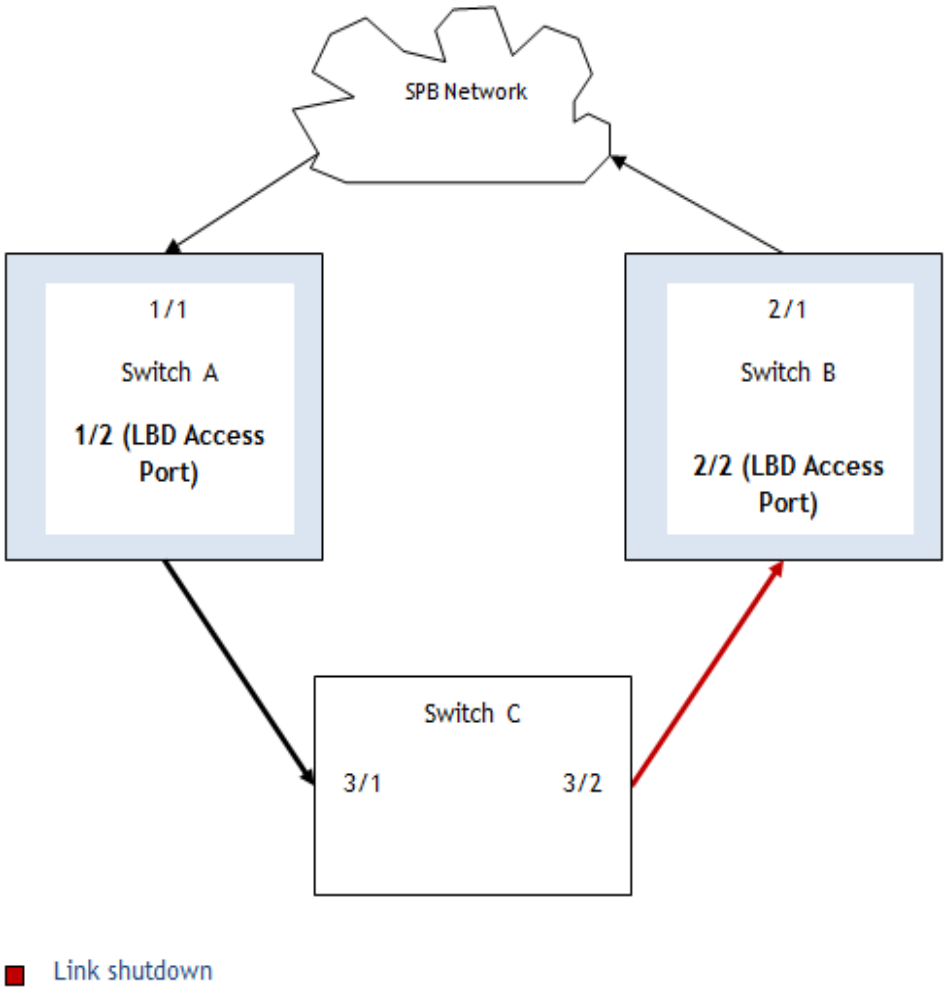


Figure 8-2 : LBD Packet Processing Mechanism for LBD Service Access Ports - Scenario 1

- Switch A and B are AOS switches running loopback-detection.
- Switch C is a legacy switch or a non AOS switch or a hub.
- 1/2 and 2/2 are SAP ports having same ISID and path cost.
- Loopback-detection is enabled with the **service-access** option on ports 1/2 and 2/2; traffic loops through 1/2 and 2/2.
- Port 2/2 is shutdown in case B has higher bridge identifier, since 1/2 and 2/2 has equal path costs.

Scenario 2

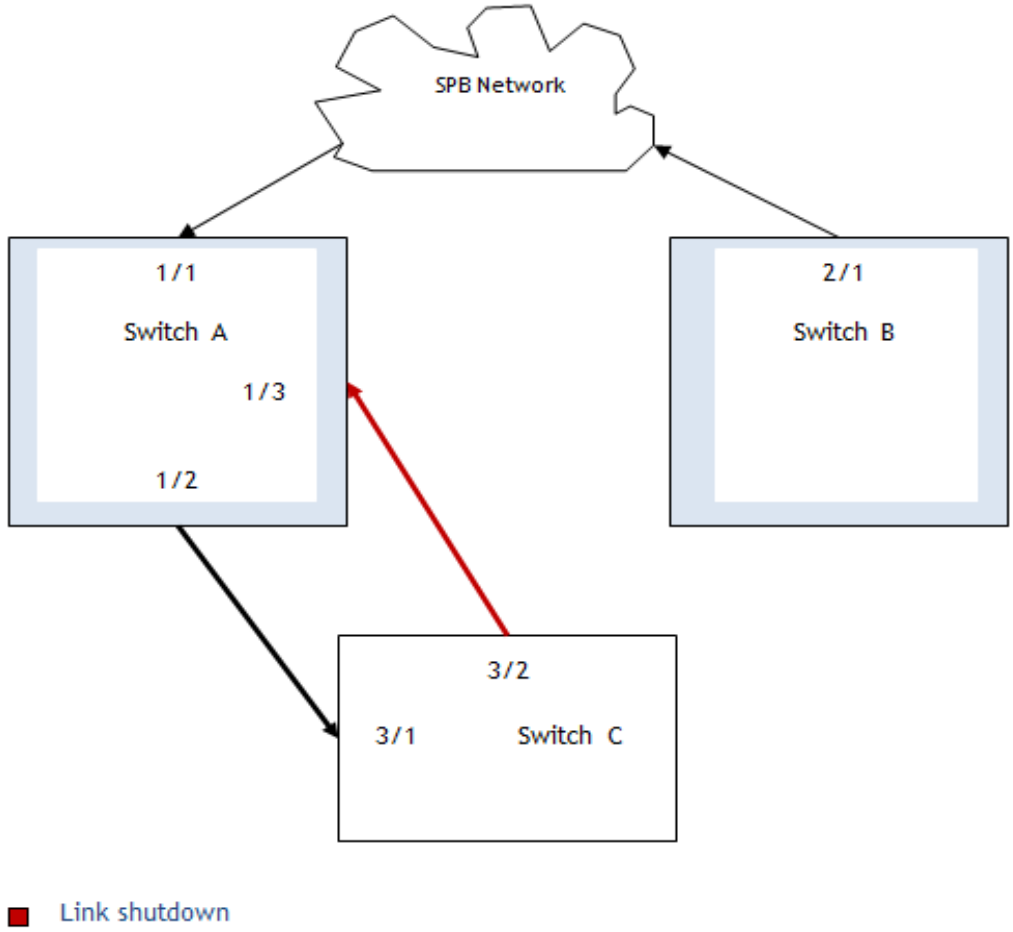


Figure 8-3 : LBD Packet Processing Mechanism for LBD Service Access Ports - Scenario 2

- Switch A and B are AOS switches running loopback-detection.
- Switch C is a legacy switch or a non AOS switch or a hub.
- 1/2 and 1/3 are SAP ports having same ISID and path cost.
- Loopback-detection is enabled with the **service-access** option on ports 1/2 and 1/3; traffic loops through 1/2 and 1/3.
- Port 1/3 is shutdown as the interface has higher port identifier, since 1/2 and 1/3 has equal path costs.

Verifying the LBD Configuration

To display LBD configuration and statistics information, use the **show** commands listed below:

loopback-detection autorecovery-timer	Displays the global LBD configuration information for the switch.
show loopback-detection port	Displays LBD configuration information for all ports on the switch.
show loopback-detection statistics port	Displays LBD statistics information for a specific port on the switch.

For more information about the resulting display from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

9 Configuring Static Link Aggregation

The OmniSwitch implementation of static link aggregation software allows you to combine several physical links into one large virtual link known as a link aggregation *group*. Using link aggregation provides the following benefits:

- **Scalability**. It is possible to configure a maximum number of link aggregation groups as mentioned in the "Network Configuration Specifications" chapter of the *OmniSwitch AOS Release 8 Specifications Guide*.
- **Reliability**. A link aggregate can operate even if one of the physical links, that is part of the link aggregate group, gets disabled.
- **Ease of Migration**. Link aggregation can ease the transition to higher bandwidth backbones.

In This Chapter

This chapter describes the basic components of static link aggregation and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- "Configuring Static Link Aggregation Groups" .
- "Adding and Deleting Ports in a Static Aggregate Group".
- "Modifying Static Aggregation Group Parameters".

Static Link Aggregation Default Values

The table below lists default values and the commands to modify them for static aggregate groups.

Parameter Description	Command	Default Value/Comments
Administrative State	**linkagg static agg admin-state**	enabled
Group Name	**linkagg static agg name**	No name configured

Quick Steps for Configuring Static Link Aggregation

Follow the steps below for a quick tutorial on configuring a static aggregate link between two switches. Additional information on how to configure each command is given in the subsections that follow.

1 Create the static aggregate link on the local switch with the **linkagg static agg size** command. For example:

```
-> linkagg static agg 1 size 4
```

2 Assign all the necessary ports with the **linkagg static port agg** command. For example:

```
-> linkagg static port 1/1-4 agg 1
```

3 Create a VLAN for this static link aggregate group with the **vlan members** command. For example:

```
-> vlan 10 members port 1
```

4 Create the equivalent static aggregate link on the *remote switch* with the **linkagg static agg size** command. For example:

```
-> linkagg static agg 1 size 4
```

5 Assign all the necessary ports with the **linkagg static port agg** command. For example:

```
-> linkagg static port 1/9-12 agg 1
```

6 Create a VLAN for this static link aggregate group with the **vlan members** command. For example:

```
-> vlan 10 members default 1
```

Note. *Optional*. You can verify your static link aggregation settings with the **show linkagg** command along with the **agg** keyword and aggregate group ID. For example:

```
 -> show linkagg agg 1

Static Aggregate
  SNMP Id                  : 40000001,
  Aggregate Number         : 1,
  SNMP Descriptor          : Omnichannel Aggregate Number 1 ref 40000001 size 4,
  Name                     : ,
  Admin State              : ENABLED,
  Operational State        : UP,
  Aggregate Size           : 4,
  Number of Selected Ports : 4,
  Number of Reserved Ports : 4,
  Number of Attached Ports : 4,
  Primary Port             : 1/1
```

You can also use the **show linkagg port** port command to display information on specific ports. See "Displaying Static Link Aggregation Configuration and Statistics" on page 9-11 for more information on the **show** commands.

An example of what these commands look like entered sequentially on the command line on the local switch:

```
-> linkagg static agg 1 size 4
-> linkagg static port 1/1-4 agg 1
-> vlan 10 port default 1
```

And an example of what these commands look like entered sequentially on the command line on the remote switch:

```
-> linkagg static agg 1 size 4
-> linkagg static port 1/9-12 agg 1
-> vlan 10 port default 1
```

Static Link Aggregation Overview

Link aggregation allows you to combine physical connections into large virtual connections known as link aggregation *groups*.

You can create Virtual LANs (VLANs), 802.1Q framing, configure Quality of Service (QoS) conditions, and other networking features on link aggregation groups because the OmnniSwitch AOS software treats these virtual links just like physical links. (See "Relationship to Other Features" on page 9-6 for more information on how link aggregation interacts with other software features.)

Load balancing for Layer 2 non-IP packets is on a MAC address basis. However when IP packets are transmitted, the balancing algorithm uses the IP address. Ports *must* be of the same speed within the same link aggregate group.

The OmniSwitch implementation of link aggregation software allows you to configure the following two different types of link aggregation groups:

- Static link aggregate groups
- Dynamic link aggregate groups

This chapter describes static link aggregation. For information on dynamic link aggregation, please refer to Chapter 10, "Configuring Dynamic Link Aggregation."

Static Link Aggregation Operation

Static link aggregate groups are virtual links between two nodes consisting multiple physical links.

Static aggregate groups can be created between OmniSwitch platforms.

Note. Static aggregate groups cannot be created between an OmniSwitch and some switches from other vendors.

The figure below shows a static aggregate group that has been configured between Switch A and Switch B. The static aggregate group links four ports on a single NI on Switch A to two ports each on separate NIs on Switch B. The network administrator has created a separate VLAN for this group so users can use this high speed link.

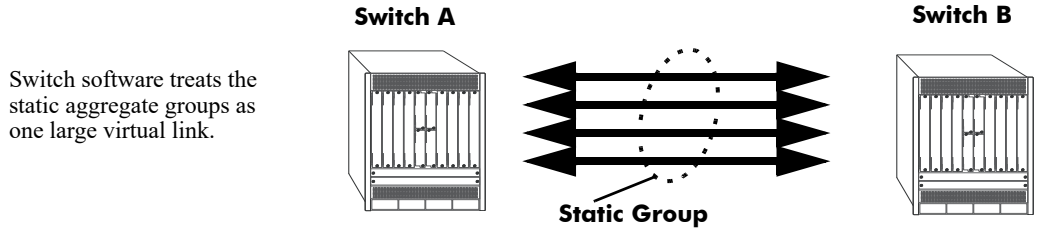


Figure 9-1 : Example of a Static Link Aggregate Group Network

See "Configuring Static Link Aggregation Groups" on page 9-6 for information on using Command Line Interface (CLI) commands to configure static aggregate groups and see "Displaying Static Link Aggregation Configuration and Statistics" on page 9-11 for information on using CLI to monitor static aggregate groups.

Relationship to Other Features

Link aggregation groups are supported by other switch software features. The following features have CLI commands or command parameters that support link aggregation:

- **VLANs**. For more information on VLANs see Chapter 4, "Configuring VLANs."
- **802.1Q**. For more information on configuring and monitoring 802.1Q see Chapter 4, "Configuring VLANs."
- **Spanning Tree**. For more information on Spanning Tree see Chapter 9, "Configuring Static Link Aggregation."

Note. See "Application Example" on page 9-10 for tutorials on using link aggregation with other features.

Configuring Static Link Aggregation Groups

This section describes how to use OmniSwitch Command Line Interface (CLI) commands to configure static link aggregate groups. See "Configuring Mandatory Static Link Aggregate Parameters" on page 9-6 for more information.

Note. See "Quick Steps for Configuring Static Link Aggregation" on page 9-3 for a brief tutorial on configuring these mandatory parameters.

The OmniSwitch implementation of link aggregation software is preconfigured with the default values for static aggregate groups as shown in the table in "Static Link Aggregation Default Values" on page 9-2. If you need to modify any of these parameters, please see "Modifying Static Aggregation Group Parameters" on page 9-9 for more information.

Note. See the "Link Aggregation Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for complete documentation of CLI commands for link aggregation.

Configuring Mandatory Static Link Aggregate Parameters

When configuring static link aggregates on a switch you must perform the following steps:

1 Create the Static Aggregate Group on the Local and Remote Switches. To create a static aggregate group use the **linkagg static agg size** command, which is described in "Creating and Deleting a Static Link Aggregate Group" on page 9-7.

2 Assign Ports on the Local and Remote Switches to the Static Aggregate Group. To assign ports to the static aggregate group you use the **linkagg static port agg** command, which is described in "Adding and Deleting Ports in a Static Aggregate Group" on page 9-7.

Note. Depending on the needs of your network you need to configure additional parameters. Commands to configure optional static aggregate parameters are described in "Modifying Static Aggregation Group Parameters" on page 9-9.

Creating and Deleting a Static Link Aggregate Group

The following subsections describe how to create and delete static link aggregate groups with the **linkagg static agg size** command.

Creating a Static Aggregate Group

To create a static aggregate group on a switch, enter **linkagg static agg** followed by the user-specified aggregate number, **size**, and the number of links in the static aggregate group:

For example, to create static aggregate group 5 that consists of eight links, on a switch, enter:

```
-> linkagg static agg 5 size 8
```

Note. The number of links assigned to a static aggregate group must always be close to the number of physical links that you plan to use. For example, if you are planning to use 2 physical links you should create a group with a size of 2 and not 4 or 8.

As an option you can also specify a name and/or the administrative status of the group by entering **linkagg static agg** followed by the user-specified aggregate number, **size**, the number of links in the static aggregate group, **name**, the optional name, **admin-state**, and either **enable** or **disable** (the default is **enable**).

For example, to create static aggregate group 5 called “static1” consisting of eight links that is administratively disabled enter:

```
-> linkagg static agg 5 size 8 name static1 admin-state disable
```

Note. If you want to specify spaces within a name for a static aggregate group the name must be specified within quotes (for example, “Static Aggregate Group 5”).

Deleting a Static Aggregate Group

To delete a static aggregation group from a switch use the **no** form of the **linkagg static agg size** command by entering **no linkagg static agg** followed by the number that identifies the group. For example, to remove static aggregate group 5 from the switch configuration, enter:

```
-> no linkagg static agg 5
```

Note. You must delete any attached ports with the **linkagg static port agg** command before you can delete a static link aggregate group.

Adding and Deleting Ports in a Static Aggregate Group

The following subsections describe how to add and delete ports in a static aggregate group with the **linkagg static port agg** command.

Adding Ports to a Static Aggregate Group

The number of ports assigned in a static aggregate group can be less than or equal to the maximum size you specified in the **linkagg static agg size** command. To assign a port to a static aggregate group you use the **linkagg static port agg** command by entering **linkagg static port** followed by the slot number, a slash

(/), the port number, **agg**, and the number or ID of the static aggregate group.

For example, to assign ports 1, 2, and 3 in slot 1 to static aggregate group 10 (which has a size of 4), enter:

```
-> linkagg static port 1/1-3 agg 10
-> linkagg static port 1/2 agg 10
-> linkagg static port 1/3 agg 10
```

Note. A port belongs to only one aggregate group.

For example, to assign port 1 in slot 1 to static aggregate group 10, enter:

```
-> linkagg static port 1/1 agg 10
```

Removing Ports from a Static Aggregate Group

To remove a port from a static aggregate group you use the **no** form of the **linkagg static port agg** command by entering **no linkagg static port** followed by the slot number, a slash (/), and the port number. For example, to remove port 4 in slot 1 from a static aggregate group, enter:

```
-> no linkagg static port 1/4
```

Ports must be deleted in the reverse order in which they were assigned. For example, if port 9 through 16 were assigned to static aggregate group 2 you must first delete port 16, then port 15, and so forth. The following is an example of how to delete ports in the proper sequence from the console:

```
-> no linkagg static port 1/24
-> no linkagg static port 1/23
-> no linkagg static port 1/22
```

Modifying Static Aggregation Group Parameters

This section describes how to modify the following static aggregate group parameters:

- Static aggregate group name (see "Modifying the Static Aggregate Group Name" on page 9-9)
- Static aggregate group administrative state (see "Modifying the Static Aggregate Group Administrative State" on page 9-9)

Modifying the Static Aggregate Group Name

The following subsections describe how to modify the name of the static aggregate group with the **linkagg static agg name** command.

Creating a Static Aggregate Group Name

To create a name for a static aggregate group by entering **linkagg static agg** followed by the number of the static aggregate group, **name**, and the user-specified name of the group. For example, to configure static aggregate group 4 with the name "Finance", enter:

```
-> linkagg static agg 4 name Finance
```

Note. If you want to specify spaces within a name for a static aggregate group the name must be specified within quotes (for example, "Static Aggregate Group 4").

Deleting a Static Aggregate Group Name

To remove a name from a static aggregate group, use the **no** form of the **linkagg static agg name** command by entering **no linkagg static agg** followed by the number of the static aggregate group and **name**. For example, to remove any user-specified name from static aggregate group 4, enter:

```
-> no linkagg static agg 4 name
```

Modifying the Static Aggregate Group Administrative State

By default, the administrative state for a static aggregate group is enabled. The following subsections describe how to enable and disable the administrative state with the **linkagg static agg admin-state** command.

Enabling the Static Aggregate Group Administrative State

To enable a static aggregate group, enter **linkagg static agg** followed by the number of the group and **admin-state enable**. For example, to enable static aggregate group 1, enter:

```
-> linkagg static agg 1 admin-state enable
```

Disabling the Static Aggregate Group Administrative State

To disable a static aggregate group by entering **linkagg static agg** followed by the number of the group and **admin-state disable**. For example, to disable static aggregate group 1, enter:

```
-> linkagg static agg 1 admin-state disable
```

Application Example

Static link aggregation groups are treated by the switch software the same way it treats individual physical ports. This section demonstrates this by providing a sample network configuration that uses static link aggregation along with other software features. In addition, a tutorial is provided that shows how to configure this sample network using Command Line Interface (CLI) commands.

The figure below shows VLAN 8, which has been configured on static aggregate 1 and uses 802.1Q tagging. The actual physical links connect ports 4/1, 4/2, 4/3, and 4/4 on Switch A to port 2/41, 2/42, 2/43, and 2/44 on Switch B.

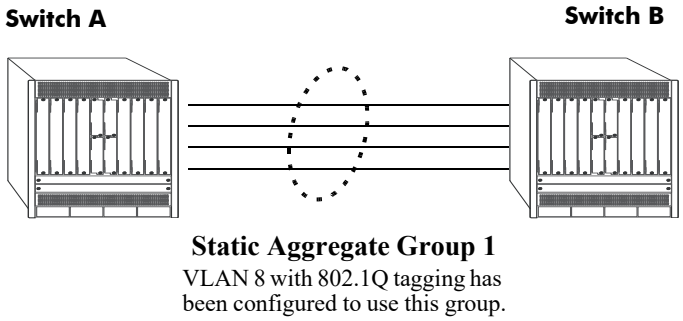


Figure 9-2 : Sample Network Using Static Link Aggregation

Follow the steps below to configure this network:

Note. Only the steps to configure the local (i.e., Switch A) switch are provided here since the steps to configure the remote (i.e., Switch B) switch are similar.

---

1 Configure static aggregate group 1 by entering **linkagg static agg 1 size 4** as shown below:

```
-> linkagg static agg 1 size 4
```

2 Assign ports 4/1, 4/2, 4/3, and 4/4 to static aggregate group 1 by entering:

```
-> linkagg static port 4/1-4 agg 1
```

3 Create VLAN 8 by entering:

```
-> vlan 8
```

4 Configure 802.1Q tagging with a tagging ID of 8 on static aggregate group 1 (on VLAN 8) by entering:

```
-> vlan 8 members linkagg 1 tagged
```

5 Repeat steps 1 through 4 on Switch B. Substitute the port numbers of the commands with the appropriate port numbers of Switch B.

Displaying Static Link Aggregation Configuration and Statistics

You can use Command Line Interface (CLI) **show** commands to display the current configuration and statistics of link aggregation. These commands include the following:

show linkagg	Displays information on link aggregation groups.
show linkagg port	Displays information on link aggregation ports.

When you use the **show linkagg** command without specifying the link aggregation group number and when you use the **show linkagg port** command without specifying the slot and port number these commands provide a "global" view of switch-wide link aggregate group and link aggregate port information, respectively.

For example, to display global statistics on all link aggregate groups (both static and dynamic), enter:

```
-> show linkagg

Number Aggregate SNMP Id  Size  Admin State Oper State Att/Sel Ports
-------+--------+--------+-----+-----------+----------+-------+-----+
1       Static   40000001  4     ENABLED     DOWN         0      0
2       Static   40000002  8     ENABLED     DOWN         0      0
10      Dynamic  40000010  8     ENABLED     DOWN         0      0
```

For example, to display global statistics on all ports associated with link aggregate groups (both static and dynamic), enter:

```
-> show linkagg port

Slot/Port Aggregate  SNMP Id  Status   Agg Oper Link Prim
---------+----------+--------+--------+---+----+----+-----
2/1        Static     2001    ATTACHED  1    UP   UP   YES
```

When you use the **show linkagg agg** command with the link aggregation group number and when you use the **show linkagg port** command with the slot and port number these commands provide detailed views of link aggregate group and link aggregate port information, respectively. These detailed views provide excellent tools for diagnosing and troubleshooting problems.

For example, to display detailed statistics for port 1 in slot 2 that is attached to static link aggregate group 1, enter:

```
-> show linkagg port 4/1

   Static Aggregable Port
    SNMP Id                       : 2001,
    Slot/Port                     : 4/1,
    Administrative State          : ENABLED,
    Operational State             : UP,
    Port State                    : ATTACHED,
    Link State                    : UP,
    Selected Agg Number           : 1,
    Port position in the aggregate: 0,
    Primary port                  : YES
```

See the "Link Aggregation Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for complete documentation of **show** commands for link aggregation.

10 Configuring Dynamic Link Aggregation

The OmniSwitch implementation of dynamic link aggregation software allows you to combine several physical links into one large virtual link known as a link aggregation *group*. Using link aggregation provides the following benefits:

- **Scalability**. It is possible to configure a maximum number of link aggregation groups as mentioned in the "Network Configuration Specifications" chapter of the *OmniSwitch AOS Release 8 Specifications Guide*.
- **Reliability**. If one of the physical links in a link aggregate group goes down (unless it is the last one) the link aggregate group can still operate.
- **Ease of Migration**. Link aggregation can ease the transition to higher bandwidth backbones.

In This Chapter

This chapter describes the basic components of dynamic link aggregation and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Configuring dynamic link aggregation groups on "Configuring Dynamic Link Aggregate Groups" on page 10-7.
- Configuring ports so they can be aggregated in dynamic link aggregation groups on "Configuring Ports to Join and Removing Ports in a Dynamic Aggregate Group" on page 10-9.
- Modifying dynamic link aggregation parameters on "Modifying Dynamic Link Aggregate Group Parameters" on page 10-11.

Dynamic Link Aggregation Default Values

The table below lists default values for dynamic aggregate groups.

Parameter Description	Command	Default Value/Comments
Group Administrative State	**linkagg lacp agg admin-state**	enabled
Group Name	**linkagg lacp agg name**	No name configured
Group Actor Administrative Key	**linkagg lacp agg actor admin-key**	0
Group Actor System Priority	**linkagg lacp agg actor system-priority**	0
Group Actor System ID	**linkagg lacp agg actor system-id**	00:00:00:00:00:00
Group Partner System ID	**linkagg lacp agg partner system-id**	00:00:00:00:00:00
Group Partner System Priority	**linkagg lacp agg partner system-priority**	0
Group Partner Administrative Key	**linkagg lacp agg partner admin-key**	0
Actor Port Administrative State	**linkagg lacp agg admin-state**	active timeout aggregate
Actor Port System ID	**linkagg lacp agg actor system-id**	00:00:00:00:00:00
Partner Port System Administrative State	**linkagg lacp agg partner admin-state**	active timeout aggregate
Partner Port Admin System ID	**linkagg lacp port partner admin system-priority**	00:00:00:00:00:00
Partner Port Administrative Key	**linkagg lacp agg partner admin-key**	0
Partner Port Admin System Priority	**linkagg lacp agg partner system-priority**	0
Actor Port Priority	**linkagg lacp port actor port priority**	0
Partner Port Administrative Port	**linkagg lacp port partner admin-port**	0
Partner Port Priority	**linkagg lacp port partner admin port-priority**	0

Quick Steps for Configuring Dynamic Link Aggregation

Follow the steps below for a quick tutorial on configuring a dynamic aggregate link between two switches. Additional information on how to configure each command is given in the subsections that follow.

1 Create the dynamic aggregate group on the local (actor) switch with the **linkagg lacp agg size** command as shown below:

```
-> linkagg lacp agg 2 size 8 actor admin-key 5
```

2 Configure ports (the number of ports must be less than or equal to the size value set in step 1) with the same actor administrative key (which allows them to be aggregated) with the **linkagg lacp agg actor admin-key** command. For example:

```
-> linkagg lacp port 1/1 actor admin-key 5
-> linkagg lacp port 1/4 actor admin-key 5
-> linkagg lacp port 3/3 actor admin-key 5
-> linkagg lacp port 5/4 actor admin-key 5
-> linkagg lacp port 6/1-2 actor admin-key 5
-> linkagg lacp port 7/3 actor admin-key 5
-> linkagg lacp port 8/1 actor admin-key 5
```

3 Create a VLAN for this dynamic link aggregate group with the **vlan** command. For example:

```
-> vlan 2 members port 2/3 untagged
```

4 Create the equivalent dynamic aggregate group on the remote (partner) switch with the **linkagg lacp agg size** command as shown below:

```
-> linkagg lacp agg 2 size 8 actor admin-key 5
```

5 Configure ports (the number of ports must be less than or equal to the size value set in step 4) with the same actor administrative key (which allows them to be aggregated) with the **linkagg lacp agg actor admin-key** command. For example:

```
-> linkagg lacp port 2/1 actor admin-key 5
-> linkagg lacp port 3/1 actor admin-key 5
-> linkagg lacp port 3/3 actor admin-key 5
-> linkagg lacp port 3/6 actor admin-key 5
-> linkagg lacp port 5/1 actor admin-key 5
-> linkagg lacp port 5/6 actor admin-key 5
-> linkagg lacp port 8/1 actor admin-key 5
-> linkagg lacp port 8/3 actor admin-key 5
```

6 Create a VLAN for this dynamic link aggregate group with the **vlan** command. For example:

```
-> vlan 2 members linkagg 2
```

Note. As an option, you can verify your dynamic aggregation group settings with the **show linkagg** command on either the actor or the partner switch. For example:

```
-> show linkagg agg 2
Dynamic Aggregate
  SNMP Id                  : 40000002,
  Aggregate Number         : 2,
  SNMP Descriptor          : Dynamic Aggregate Number 2 ref 40000002 size 8,
  Name                     : ,
  Admin State              : ENABLED,
  Operational State        : UP,
  Aggregate Size           : 8,
  Number of Selected Ports : 8,
  Number of Reserved Ports : 8,
  Number of Attached Ports : 8,
  Primary Port             : 1/1,
LACP
  MACAddress               : [00:1f:cc:00:00:00],
  Actor System Id          : [00:20:da:81:d5:b0],
  Actor System Priority    : 0,
  Actor Admin Key          : 5,
  Actor Oper Key           : 0,
  Partner System Id        : [00:20:da:81:d5:b1],
  Partner System Priority  : 0,
  Partner Admin Key        : 5,
  Partner Oper Key         : 0
```

You can also use the **show linkagg port** port command to display information on specific ports. See "Displaying Dynamic Link Aggregation Configuration and Statistics" on page 10-28 for more information on **show** commands.

An example of what these commands look like entered sequentially on the actor switch command line:

```
-> linkagg lacp agg 2 size 8 actor admin-key 5
-> linkagg lacp port 1/1 actor admin-key 5
-> linkagg lacp port 1/4 actor admin-key 5
-> linkagg lacp port 3/3 actor admin-key 5
-> linkagg lacp port 5/4 actor admin-key 5
-> linkagg lacp port 6/1-2 actor admin-key 5
-> linkagg lacp port 7/3 actor admin-key 5
-> linkagg lacp port 8/1 actor admin-key 5
-> vlan 2 port default 2
```

An example of what these commands look like entered sequentially on the partner switch command line:

```
-> linkagg lacp agg 2 size 8 actor admin-key 5
-> linkagg lacp port 2/1 actor admin-key 5
-> linkagg lacp port 3/1 actor admin-key 5
-> linkagg lacp port 3/3 actor admin-key 5
-> linkagg lacp port 3/6 actor admin-key 5
-> linkagg lacp port 5/1 actor admin-key 5
-> linkagg lacp port 5/6 actor admin-key 5
-> linkagg lacp port 8/1 actor admin-key 5
-> linkagg lacp port 8/3 actor admin-key 5
-> vlan 2 port default 2
```

Dynamic Link Aggregation Overview

Link aggregation allows you to combine physical connections into large virtual connections known as link aggregation *groups*.

You can create Virtual LANs (VLANs), 802.1Q framing, configure Quality of Service (QoS) conditions, and other networking features on link aggregation groups because switch software treats these virtual links just like physical links. (See "Relationship to Other Features" on page 10-7 for more information on how link aggregation interacts with other software features.)

Link aggregation groups are identified by unique MAC addresses, which are created by the switch but can be modified by the user at any time. Load balancing for Layer 2 non-IP packets is on a MAC address basis and for IP packets the balancing algorithm uses the IP address as well. Ports *must* be of the same speed within the same aggregate group.

The OmniSwitch implementation of link aggregation software allows you to configure the following two different types of link aggregation groups:

- Static link aggregate groups
- Dynamic link aggregate groups

This chapter describes dynamic link aggregation. For information on static link aggregation, please refer to Chapter 9, "Configuring Static Link Aggregation."

Dynamic Link Aggregation Operation

Dynamic aggregate groups are virtual links between two nodes consisting of physical links. Dynamic aggregate groups use the standard IEEE 802.3ad Link Aggregation Control Protocol (LACP) to dynamically establish the best possible configuration for the group. This task is accomplished by special Link Aggregation Control Protocol Data Unit (LACPDU) frames that are sent and received by switches on both sides of the link to monitor and maintain the dynamic aggregate group.

The figure on the following page shows a dynamic aggregate group that has been configured between Switch A and Switch B. The dynamic aggregate group links four ports on Switch A to four ports on Switch B.

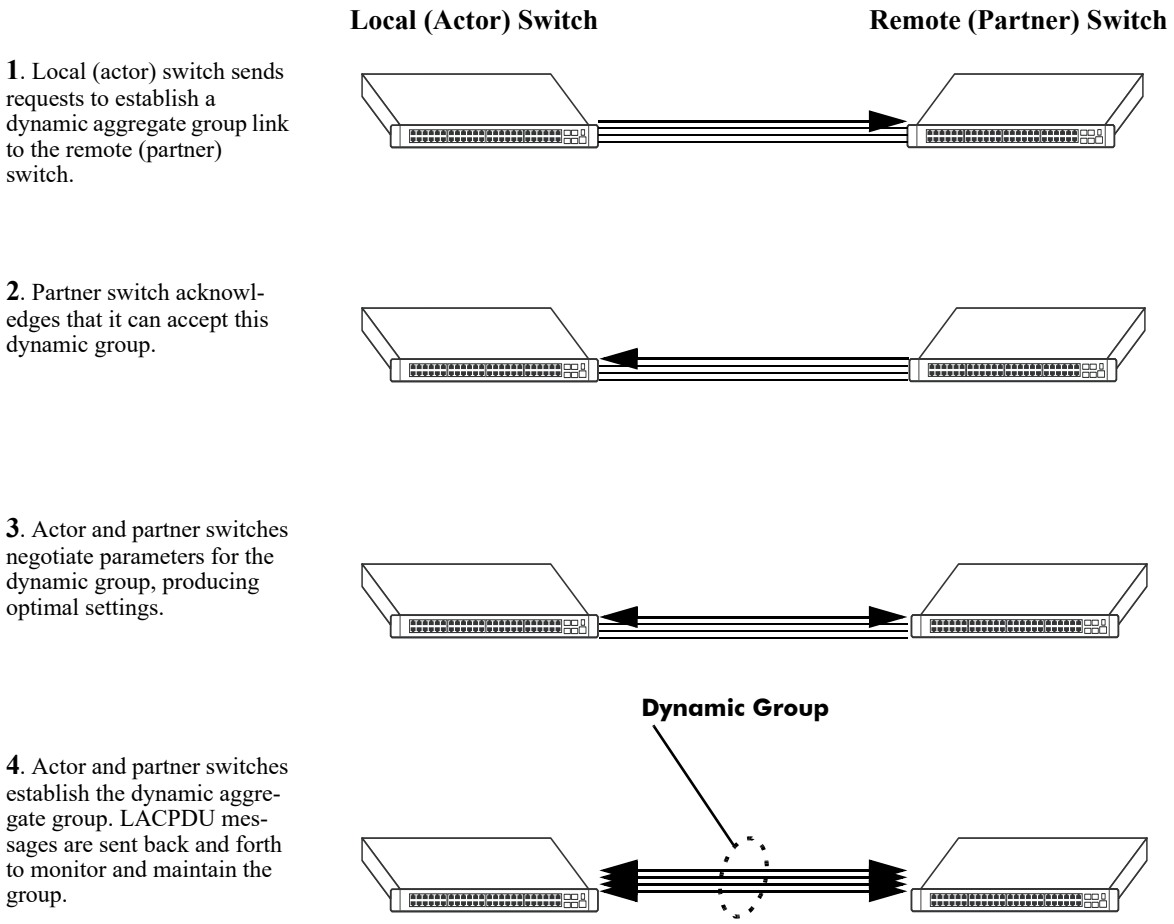


Figure 10-1 : Example of a Dynamic Aggregate Group Network

See "Configuring Dynamic Link Aggregate Groups" on page 10-7 for information on using Command Line Interface (CLI) commands to configure dynamic aggregate groups and see "Displaying Dynamic Link Aggregation Configuration and Statistics" on page 10-28 for information on using the CLI to monitor dynamic aggregate groups.

Relationship to Other Features

Link aggregation groups are supported by other switch software features. For example, you can configure 802.1Q tagging on link aggregation groups in addition to configuring it on individual ports. The following features have CLI commands or command parameters that support link aggregation:

- **VLANs**. For more information on VLANs, see Chapter 4, "Configuring VLANs."
- **802.1Q**. For more information on configuring and monitoring 802.1Q, see Chapter 4, "Configuring VLANs."
- **Spanning Tree**. For more information on Spanning Tree, see Chapter 6, "Configuring Spanning Tree Parameters."

Note. See "Application Examples" on page 10-25 for tutorials on using link aggregation with other features.

Configuring Dynamic Link Aggregate Groups

This section describes how to use Command Line Interface (CLI) commands to create, modify, and delete dynamic aggregate groups. See "Configuring Mandatory Dynamic Link Aggregate Parameters" on page 10-8 for more information.

Note. See "Quick Steps for Configuring Dynamic Link Aggregation" on page 10-3 for a brief tutorial on configuring these mandatory parameters.

The OmniSwitch implementation of link aggregation software is preconfigured with the default values for dynamic aggregate groups and ports shown in the table in "Dynamic Link Aggregation Default Values" on page 10-2. For most configurations, using only the steps described in "Creating and Deleting a Dynamic Aggregate Group" on page 10-8 is necessary to configure a dynamic link aggregate group. However, if you need to modify any of the parameters listed in the table on page 10-2, please see "Modifying Dynamic Link Aggregate Group Parameters" on page 10-11 for more information.

Note. See the "Link Aggregation Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for complete documentation of **show** commands for link aggregation.

Configuring Mandatory Dynamic Link Aggregate Parameters

When configuring LACP link aggregates on a switch you must perform the following steps:

1 Create the Dynamic Aggregate Groups on the Local (Actor) and Remote (Partner) Switches. To create a dynamic aggregate group use the **linkagg lacp agg size** command, which is described in "Creating and Deleting a Dynamic Aggregate Group" on page 10-8.

2 Configure the Same Administrative Key on the Ports You Want to Join the Dynamic Aggregate Group. To configure ports with the same administrative key (which allows them to be aggregated), use the **linkagg lacp agg actor admin-key** command, which is described in "Configuring Ports to Join and Removing Ports in a Dynamic Aggregate Group" on page 10-9.

Note. Depending on the needs of your network you need to configure additional parameters. Commands to configure optional dynamic link aggregate parameters are described in "Modifying Dynamic Link Aggregate Group Parameters" on page 10-11.These commands must be executed after you create a dynamic aggregate group.

Creating and Deleting a Dynamic Aggregate Group

The following subsections describe how to create and delete dynamic aggregate groups with the **linkagg lacp agg size** command.

Creating a Dynamic Aggregate Group

To configure a dynamic aggregate group, enter **linkagg lacp agg** followed by the user-configured dynamic aggregate number, **size**, and the maximum number of links that belong to this dynamic aggregate group. For example, to configure the dynamic aggregate group 2 consisting of eight links enter:

```
-> linkagg lacp agg 2 size 8
```

You can create link aggregation (both static and dynamic) groups for a standalone or a chassis-based switch. In addition, you can also specify optional parameters shown in the table below. These parameters must be entered after **size** and the user-specified number of links.

linkagg lacp agg size keywords		
name	**admin state enable**	**partner admin-key**
actor system-priority	**admin state disable**	**actor admin-key**
partner system-priority	**actor system-id**	**partner system-id**

For example, assigning the actor admin key when you create the dynamic aggregate group is recommended to help ensure that ports are assigned to the correct group. To create a dynamic aggregate group with aggregate number 3 consisting of two ports with an admin actor key of 10, for example, enter:

```
-> linkagg lacp agg 3 size 2 actor admin-key 10
```

Note. The optional keywords for this command can be entered in any order as long as they are entered after **size** and the user-specified number of links.

Deleting a Dynamic Aggregate Group

To remove a dynamic aggregation group configuration from a switch use the **no** form of the **linkagg lacp agg size** command by entering **no linkagg lacp agg** followed by its dynamic aggregate group number.

For example, to delete dynamic aggregate group 2 from the switch configuration, enter:

```
-> no linkagg lacp agg 2
```

Note. You cannot delete a dynamic aggregate group if it has any attached ports. To remove attached ports you must disable the dynamic aggregate group with the **linkagg lacp agg admin-state** command, which is described in "Disabling a Dynamic Aggregate Group" on page 10-12.

Configuring Ports to Join and Removing Ports in a Dynamic Aggregate Group

The following subsections describe how to configure ports with the same administrative key (which allows them to be aggregated) or to remove them from a dynamic aggregate group with the **linkagg lacp agg actor admin-key** command.

Configuring Ports To Join a Dynamic Aggregate Group

To configure ports with the same administrative key (which allows them to be aggregated) enter **lacp port** followed by the slot number, a slash (/), the port number, **actor admin-key**, and the user-specified actor administrative key. Ports must be of the same speed.

For example, to configure ports 1, 2, and 3 in slot 4 with an administrative key of 10, enter:

```
-> linkagg lacp port 4/1-3 actor admin-key 10
```

Note. A port can belong to only one aggregate group.

You must execute the **linkagg lacp port actor admin-key** command on all ports in a dynamic aggregate group. If not, the ports are unable to join the group.

In addition, you can also specify optional parameters shown in the table below. These keywords must be entered after the actor admin-key and the user-specified actor administrative key value.

lacp agg actor admin-key keywords		
actor admin-state	**partner admin-state**	**actor system-id**
actor system-priority	**partner admin system-id**	**partner admin-key**
partner admin-system-priority	**actor port-priority**	**partner admin-port**
partner admin port-priority		

Note. The **actor admin-state** and **partner admin-state** keywords have additional parameters, which are described in "Modifying the Actor Port System Administrative State" on page 10-16 and "Modifying the Partner Port System Administrative State" on page 10-20, respectively.

All of the optional keywords listed above for this command can be entered in any order as long as they appear after the **actor admin-key** keywords and their user-specified value.

For example, to configure actor administrative key of 10, a local system ID (MAC address) of 00:20:da:06:ba:d3, and a local priority of 65535 to slot 4 port 1, enter:

```
-> linkagg lacp port 4/1 actor admin-key 10 actor system-id 00:20:da:06:ba:d3
actor system-priority 65535
```

For example, to configure an actor administrative key of 10 to slot 4 port 1, enter:

```
-> linkagg lacp port 4/1 actor admin-key 10
```

Removing Ports from a Dynamic Aggregate Group

To remove a port from a dynamic aggregate group, use the **no** form of the **linkagg lacp agg actor admin-key** command by entering **linkagg lacp port** followed by the slot number, a slash (/), and the port number.

For example, to remove port 4 in slot 4 from any dynamic aggregate group, enter:

```
-> no linkagg lacp port 4/4
```

Ports must be deleted in the reverse order in which they were configured. For example, if port 4/4 through 4/6 were configured to join dynamic aggregate group 2 you must first delete port 4/6, then port 4/5, and so forth. The following is an example of how to delete ports in the proper sequence from the console:

```
-> no linkagg lacp port 4/6
-> no linkagg lacp port 4/5
-> no linkagg lacp port 4/4
```

Modifying Dynamic Link Aggregate Group Parameters

The table on page 10-2 lists default group and port settings for the OmniSwitch implementation of Dynamic Link Aggregation. These parameters ensure compliance with the IEEE 802.3ad specification. For most networks, these default values need not be modified or can be modified automatically by the switch software. However, if you need to modify any of these default settings, see the following sections to modify the parameters for:

- Dynamic aggregate groups on page 10-11
- Dynamic aggregate actor ports on page 10-16
- Dynamic aggregate partner ports on page 10-19

Note. You *must* create a dynamic aggregate group before you can modify group or port parameters. See "Configuring Dynamic Link Aggregate Groups" on page 10-7 for more information.

Modifying Dynamic Aggregate Group Parameters

This section describes how to modify the following dynamic aggregate group parameters:

- Group name (see "Modifying the Dynamic Aggregate Group Name" on page 10-12)
- Group administrative state (see "Modifying the Dynamic Aggregate Group Administrative State" on page 10-12)
- Group local (actor) switch actor administrative key (see "Configuring and Deleting the Dynamic Aggregate Group Actor Administrative Key" on page 10-13)
- Group local (actor) switch system priority (see "Modifying the Dynamic Aggregate Group Actor System Priority" on page 10-13)
- Group local (actor) switch system ID (see "Modifying the Dynamic Aggregate Group Actor System ID" on page 10-14)
- Group remote (partner) administrative key (see "Modifying the Dynamic Aggregate Group Partner Administrative Key" on page 10-14)
- Group remote (partner) system priority (see "Modifying the Dynamic Aggregate Group Partner System Priority" on page 10-15)
- Group remote (partner) switch system ID (see "Modifying the Dynamic Aggregate Group Partner System ID" on page 10-15)

Modifying the Dynamic Aggregate Group Name

The following subsections describe how to configure and remove a dynamic aggregate group name with the **linkagg lacp agg name** command.

Configuring a Dynamic Aggregate Group name

To configure a dynamic aggregate group name, enter **linkagg lacp agg** followed by the dynamic aggregate group number, **name**, and the user-specified name.

For example, to name dynamic aggregate group 4 “Engineering”, enter:

```
-> linkagg lacp agg 4 name Engineering
```

Note. If you want to specify spaces within a name, the name must be enclosed in quotes. For example:

```
-> linkagg lacp agg 4 name "Engineering Lab"
```

Deleting a Dynamic Aggregate Group Name

To remove a dynamic aggregate group name from the configuration of a switch, use the **no** form of the **linkagg lacp agg name** command by entering **linkagg lacp agg** followed by the dynamic aggregate group number and **no name**.

For example, to remove any user-configured name from dynamic aggregate group 4, enter:

```
-> no linkagg lacp agg 4 name
```

Modifying the Dynamic Aggregate Group Administrative State

By default, the dynamic aggregate group administrative state is enabled. The following subsections describe how to enable and disable the administrative state of a dynamic aggregate group with the **linkagg lacp agg admin-state** command.

Enabling a Dynamic Aggregate Group

To enable the dynamic aggregate group administrative state, enter **linkagg lacp agg** followed by the dynamic aggregate group number and **admin-state enable**. For example, to enable dynamic aggregate group 4, enter:

```
-> linkagg lacp agg 4 admin-state enable
```

Disabling a Dynamic Aggregate Group

To disable the administrative state of a dynamic aggregate group, use the **linkagg lacp agg admin-state** command by entering **linkagg lacp agg** followed by the dynamic aggregate group number and **admin-state disable**.

For example, to disable dynamic aggregate group 4, enter:

```
-> linkagg lacp agg 4 admin-state disable
```

Configuring and Deleting the Dynamic Aggregate Group Actor Administrative Key

The following subsections describe how to configure and delete a dynamic aggregate group actor administrative key with the **linkagg lacp agg actor admin-key** command.

Configuring a Dynamic Aggregate Actor Administrative Key

To configure the dynamic aggregate group actor switch administrative key, enter **linkagg lacp agg** followed by the dynamic aggregate group number, **actor admin-key**, and the value for the administrative key.

For example, to configure dynamic aggregate group 4 with an administrative key of 10, enter:

```
-> linkagg lacp agg 4 actor admin-key 10
```

Deleting a Dynamic Aggregate Actor Administrative Key

To remove an actor switch administrative key from the configuration of a dynamic aggregate group, use the **no** form of the **linkagg lacp agg actor admin-key** command by entering **linkagg lacp agg** followed by the dynamic aggregate group number and the **actor admin-key** parameter.

For example, to remove an administrative key from dynamic aggregate group 4, enter:

```
-> no linkagg lacp agg 4 actor admin-key
```

Modifying the Dynamic Aggregate Group Actor System Priority

By default, the dynamic aggregate group actor system priority is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp agg actor system-priority** command.

Configuring a Dynamic Aggregate Group Actor System Priority

You can configure a user-specified dynamic aggregate group actor system priority value by entering **linkagg lacp agg** followed by the dynamic aggregate group number, **actor system-priority**, and the new priority value.

For example, to change the actor system priority of dynamic aggregate group 4 to 2000, enter:

```
-> linkagg lacp agg 4 actor system-priority 2000
```

Restoring the Dynamic Aggregate Group Actor System Priority

To restore the dynamic aggregate group actor system priority to its default (0) value use the **no** form of the **linkagg lacp agg actor system-priority** command by entering **no linkagg lacp agg** followed by the dynamic aggregate group number and **no actor system priority**.

For example, to restore the actor system priority to its default value on dynamic aggregate group 4, enter:

```
-> no linkagg lacp agg 4 actor system-priority
```

Modifying the Dynamic Aggregate Group Actor System ID

By default, the dynamic aggregate group actor system ID (MAC address) is 00:00:00:00:00:00. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp agg actor system-id** command.

Configuring a Dynamic Aggregate Group Actor System ID

You can configure a user-specified dynamic aggregate group actor system ID by entering **linkagg lacp agg** followed by the dynamic aggregate group number, **actor system-id**, and the user-specified MAC address (in the hexadecimal format of *xx:xx:xx:xx:xx:xx*), which is used as the system ID.

For example, to configure the system ID on dynamic aggregate group 4 as 00:20:da:81:d5:b0, enter:

```
-> linkagg lacp agg 4 actor system-id 00:20:da:81:d5:b0
```

Restoring the Dynamic Aggregate Group Actor System ID

To remove the user-configured actor switch system ID from the configuration of a dynamic aggregate group, use the **no** form of the **linkagg lacp agg actor system-id** command by entering **linkagg lacp agg** followed by the dynamic aggregate group number and **actor system-id**.

For example, to remove the user-configured system ID from dynamic aggregate group 4, enter:

```
-> no linkagg lacp agg 4 actor system-id
```

Modifying the Dynamic Aggregate Group Partner Administrative Key

By default, the dynamic aggregate group partner administrative key (the administrative key of the partner switch) is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp agg partner admin-key** command.

Configuring a Dynamic Aggregate Group Partner Administrative Key

You can modify the dynamic aggregate group partner administrative key to a value ranging from 0 to 65535 by entering **linkagg lacp agg** followed by the dynamic aggregate group number, **partner admin-key** parameter, and the value for the administrative key.

For example, to set the partner administrative key to 4 on dynamic aggregate group 4, enter:

```
-> linkagg lacp agg 4 partner admin-key 10
```

Restoring the Dynamic Aggregate Group Partner Administrative Key

To remove a partner administrative key from the configuration of a dynamic aggregate group, use the **no** form of the **linkagg lacp agg partner admin-key** command by entering **no linkagg lacp agg** followed by the dynamic aggregate group number and the **partner admin-key** parameter.

For example, to remove the user-configured partner administrative key from dynamic aggregate group 4, enter:

```
-> no linkagg lacp agg 4 partner admin-key
```

Modifying the Dynamic Aggregate Group Partner System Priority

By default, the dynamic aggregate group partner system priority is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp agg partner system-priority** command.

Configuring a Dynamic Aggregate Group Partner System Priority

You can modify the dynamic aggregate group partner system priority to a value by entering **linkagg lacp agg** followed by the dynamic aggregate group number, **partner system-priority**, and the new priority value.

For example, to set the partner system priority on dynamic aggregate group 4 to 2000, enter:

```
-> linkagg lacp agg 4 partner system-priority 2000
```

Restoring the Dynamic Aggregate Group Partner System Priority

To restore the dynamic aggregate group partner system priority to its default (0) value use the **no** form of the **linkagg lacp agg partner system-priority** command by entering **no linkagg lacp agg** followed by the dynamic aggregate group number and **partner system-priority**.

For example, to reset the partner system priority of dynamic aggregate group 4 to its default value, enter:

```
-> no linkagg lacp agg 4 partner system-priority
```

Modifying the Dynamic Aggregate Group Partner System ID

By default, the dynamic aggregate group partner system ID is 00:00:00:00:00:00. The following subsections describe how to configure a user-specified value and how to restore it to its default value with the **linkagg lacp agg partner system-id** command.

Configuring a Dynamic Aggregate Group Partner System ID

You can configure the dynamic aggregate group partner system ID by entering **linkagg lacp agg** followed by the dynamic aggregate group number, **partner system-id**, and the user-specified MAC address (in the hexadecimal format of *xx:xx:xx:xx:xx:xx*), which is used as the system ID.

For example, to configure the partner system ID as 00:20:da:81:d5:b0 on dynamic aggregate group 4, enter:

```
-> linkagg lacp agg 4 partner system-id 00:20:da:81:d5:b0
```

Restoring the Dynamic Aggregate Group Partner System ID

To remove the user-configured partner switch system ID from the configuration of a dynamic aggregate group, use the **no** form of the **linkagg lacp agg partner system-id** command by entering **no linkagg lacp agg** followed by the dynamic aggregate group number and the **partner system-id** parameter.

For example, to remove the user-configured partner system ID from dynamic aggregate group 4, enter:

```
-> no linkagg lacp agg 4 partner system-id
```

Modifying Dynamic Link Aggregate Actor Port Parameters

This section describes how to modify the following dynamic aggregate actor port parameters:

- Actor port administrative state (see "Modifying the Actor Port System Administrative State" on page 10-16)
- Actor port system ID (see "Modifying the Actor Port System ID" on page 10-17)
- Actor port system priority (see "Modifying the Actor Port System Priority" on page 10-18)
- Actor port priority (see "Modifying the Actor Port Priority" on page 10-19)

Notes.

- See "Configuring Ports to Join and Removing Ports in a Dynamic Aggregate Group" on page 10-9 for information on modifying a dynamic aggregate group administrative key.
- A port can belong to only one aggregate group.

Modifying the Actor Port System Administrative State

The system administrative state of a dynamic aggregate group actor port is indicated by bit settings in Link Aggregation Control Protocol Data Unit (LACPDU) frames sent by the port. By default, bits 0 (indicate that the port is active), 1 (indicate that short timeouts are used for LACPDU frames), and 2 (indicate that this port is available for aggregation) are set in LACPDU frames.

The following subsections describe how to configure user-specified values and how to restore them to their default values with the **linkagg lacp agg admin-state** command.

Configuring Actor Port Administrative State Values

To configure the system administrative state values of the LACP actor port, enter **linkagg lacp port**, the slot number, a slash (/), the port number, **actor admin-state,** and one or more of the keywords shown in the table below *or* use the **none** keyword:

linkagg lacp agg actor admin-state Keyword	Definition
active	Specifies that bit 0 in LACPDU frames is set, which indicates that the link is able to exchange LACPDU frames. By default, this bit is set.
timeout	Specifies that bit 1 in LACPDU frames is set, which indicates that a short time-out is used for LACPDU frames. When this bit is disabled, a long time-out is used for LACPDU frames. By default, this bit is set.
aggregate	Specifies that bit 2 in LACPDU frames is set, which indicates that the system considers this link to be a potential candidate for aggregation. If this bit is not set, the system considers the link to be individual (it can only operate as a single link). By default, this bit is set.
synchronize	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 3) is set by the system, the port is allocated to the correct dynamic aggregation group. If this bit is not set by the system, the port is not allocated to the correct dynamic aggregation group.

linkagg lacp agg actor admin-state Keyword	Definition
collect	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 4) is set by the system, incoming LACPDU frames are collected from the individual ports that make up the dynamic aggregate group.
distribute	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 5) is set by the system, distributing outgoing frames on the port is disabled.
default	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 6) is set by the system, it indicates that the actor is using defaulted partner information administratively configured for the partner.
expire	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 7) is set by the system, the actor cannot receive LACPDU frames.

Note. Specifying **none** removes all administrative states from the LACPDU configuration. For example:

```
-> linkagg lacp port 5/49 actor admin-state none
```

For example, to set bits 0 (**active**) and 2 (**aggregate**) on dynamic aggregate actor port 49 in slot 5 you would enter:

```
-> linkagg lacp port 5/49 actor admin-state active aggregate
```

For example, to set bits 0 (**active**) and 2 (**aggregate**) on dynamic aggregate actor port 49 in slot 5, enter:

```
-> linkagg lacp port 5/49 actor admin-state active aggregate
```

Restoring Actor Port Administrative State Values

To restore LACPDU bit settings to their default values, use the **no** form of the **linkagg lacp port actor admin-state** command by entering the **active**, **timeout**, and **aggregate** keywords.

For example, to restore bits 0 (**active**) and 2 (**aggregate**) to their default settings on dynamic aggregate actor port 2 in slot 5, enter:

```
-> no linkagg lacp port 5/2 actor admin-state active aggregate
```

Note. Since individual bits with the LACPDU frame are set with the **linkagg lacp agg actor admin-state** command you can set some bits on and restore other bits within the same command. For example, if you wanted to restore bit 2 (**aggregate**) to its default settings and set bit 0 (**active**) on dynamic aggregate actor port 49 in slot 5 you would enter:

```
-> no linkagg lacp agg 5/49 actor admin-state active aggregate
```

Modifying the Actor Port System ID

By default, the actor port system ID (the MAC address used as the system ID on dynamic aggregate actor ports) is 00:00:00:00:00:00. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp port actor system-id** command.

Configuring an Actor Port System ID

You can configure the actor port system ID by entering **linkagg lacp port**, the slot number, a slash (/), the port number, **actor system-id**, and the user specified actor port system ID (MAC address) in the hexadecimal format of xx:xx:xx:xx:xx:xx.

For example, to modify the system ID of the dynamic aggregate actor port 3 in slot 7 to **00:20:da:06:ba:d3**, enter:

```
-> linkagg lacp port 7/3 actor system-id 00:20:da:06:ba:d3
```

For example, to modify the system ID of the dynamic aggregate actor port 3 in slot 7 to **00:20:da:06:ba:d3** and document that the port is 10 Mbps Ethernet you would enter:

```
-> linkagg lacp port 7/3 actor system-id 00:20:da:06:ba:d3
```

Restoring the Actor Port System ID

To remove a user-configured system ID from a dynamic aggregate group actor port configuration, use the **no** form of the **linkagg lacp port actor system-id** command by entering **no linkagg lacp agg**, the slot number, a slash (/), the port number, and **actor system-id** keyword.

For example, to remove a user-configured system ID from dynamic aggregate actor port 3 in slot 7, enter:

```
-> linkagg lacp port 7/3 actor system-id
```

Modifying the Actor Port System Priority

By default, the actor system priority is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp port actor system-priority** command.

Configuring an Actor Port System Priority

You can configure the actor system priority to a value by entering **lacp agg**, the slot number, a slash (/), the port number, **actor system priority**, and the user-specified actor port system priority.

For example, to modify the system priority of dynamic aggregate actor port 5 in slot 2 to 200 you would enter:

```
-> linkagg lacp port 2/5 actor system-priority 200
```

For example, to modify the system priority of dynamic aggregate actor port 5 in slot 2 to 200, enter:

```
-> linkagg lacp port 2/5 actor system-priority 200
```

Restoring the Actor Port System Priority

To remove a user-configured actor port system priority from a dynamic aggregate group actor port configuration use the **no** form of the **linkagg lacp port actor system-priority** command by entering **no linkagg lacp agg**, the slot number, a slash (/), the port number, and **actor system priority**.

For example, to remove a user-configured system priority from dynamic aggregate actor port 5 in slot 2 you would enter:

```
-> no linkagg lacp port 2/5 actor system-priority
```

Modifying the Actor Port Priority

By default, the actor port priority (used to converge dynamic key changes) is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp port actor port priority** command.

Configuring the Actor Port Priority

You can configure the actor port priority to a value by entering **linkagg lacp agg**, the slot number, a slash (/), the port number, **actor port-priority**, and the user-specified actor port priority.

For example, to modify the actor port priority of dynamic aggregate actor port 1 in slot 2 to 100 you would enter:

```
-> linkagg lacp port 2/1 actor port-priority 100
```

For example, to modify the actor port priority of dynamic aggregate actor port 1 in slot 2 to 100, enter:

```
-> linkagg lacp port 2/1 actor port-priority 100
```

Restoring the Actor Port Priority

To remove a user configured actor port priority from a dynamic aggregate group actor port configuration use the **no** form of the **linkagg lacp port actor port priority** command by entering **no linkagg lacp agg**, the slot number, a slash (/), the port number, and **no actor port priority**.

For example, to remove a user-configured actor priority from dynamic aggregate actor port 1 in slot 2 you would enter:

```
-> no linkagg lacp port 2/1 actor port-priority
```

Modifying Dynamic Aggregate Partner Port Parameters

This section describes how to modify the following dynamic aggregate partner port parameters:

- Partner port system administrative state (see "Modifying the Partner Port System Administrative State" on page 10-20)
- Partner port administrative key (see "Modifying the Partner Port Administrative Key" on page 10-21)
- Partner port system ID (see "Modifying the Partner Port System ID" on page 10-22)
- Partner port system priority (see "Modifying the Partner Port System Priority" on page 10-22)
- Partner port administrative state (see "Modifying the Partner Port Administrative Status" on page 10-23)
- Partner port priority (see "Modifying the Partner Port Priority" on page 10-23)

See Chapter 1, "Configuring Ethernet Ports," for information on configuring Ethernet ports.

Note. A port can belong to only one aggregate group.

Modifying the Partner Port System Administrative State

The system administrative state of a dynamic aggregate group partner (remote switch) port is indicated by bit settings in Link Aggregation Control Protocol Data Unit (LACPDU) frames sent by this port. By default, bits 0 (indicating that the port is active), 1 (indicating that short timeouts are used for LACPDU frames), and 2 (indicating that this port is available for aggregation) are set in LACPDU frames.

The following subsections describe how to configure user-specified values and how to restore them to their default values with the **linkagg lacp agg partner admin-state** command.

Configuring Partner Port System Administrative State Values

To configure the system administrative state values for the port on the dynamic aggregate partner, enter **linkagg lacp port**, the slot number, a slash (/), the port number, **partner admin-state,** and one or more of the keywords shown in the table below *or* **none**:

Keyword	Definition
active	Specifies that bit 0 in LACPDU frames is set, which indicates that the link is able to exchange LACPDU frames. By default, this bit is set.
timeout	Specifies that bit 1 in LACPDU frames is set, which indicates that a short time-out is used for LACPDU frames. When this bit is disabled, a long time-out is used for LACPDU frames. By default, this bit is set.
aggregate	Specifies that bit 2 in LACPDU frames is set, which indicates that the system considers this link to be a potential candidate for aggregation. If this bit is not set, the system considers the link to be individual (it can only operate as a single link). By default, this bit is set.
synchronize	Specifies that bit 3 in the partner state octet is enabled. When this bit is set, the port is allocated to the correct dynamic aggregation group. If this bit is not enabled, the port is not allocated to the correct aggregation group. By default, this value is disabled.
collect	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 4) is set by the system, incoming LACPDU frames are collected from the individual ports that make up the dynamic aggregate group.
distribute	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 5) is set by the system, distributing outgoing frames on the port is disabled.
default	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 6) is set by the system, it indicates that the partner is using defaulted actor information administratively configured for the partner.
expire	Specifying this keyword has no effect because the system always determines its value. When this bit (bit 7) is set by the system, the actor cannot receive LACPDU frames.

Note. Specifying **none** removes all administrative states from the LACPDU configuration. For example:

```
-> linkagg lacp port 7/49 partner admin-state none
```

For example, to set bits 0 (**active**) and 2 (**aggregate**) on dynamic aggregate partner port 49 in slot 7, enter:

```
-> linkagg lacp port 7/49 partner admin-state active aggregate
```

For example, to set bits 0 (**active**) and 2 (**aggregate**) on dynamic aggregate partner port 49 in slot 7 and document that the port is a Gigabit Ethernet port, enter:

```
-> linkagg lacp port 7/49 partner admin-state active aggregate
```

Restoring Partner Port System Administrative State Values

To restore LACPDU bit settings to their default values use the **no** form of the **linkagg lacp agg partner admin-state** command and enter the **active**, **timeout**, **aggregate**, or **synchronize** keywords.

For example, to restore bits 0 (**active**) and 2 (**aggregate**) to their default settings on dynamic aggregate partner port 1 in slot 7, enter:

```
-> no linkagg lacp port 7/1 partner admin-state active aggregate
```

Note. Since individual bits with the LACPDU frame are set with the **linkagg lacp port partner admin state** command you can set some bits on and restore other bits to default values within the same command. For example, if you wanted to restore bit 2 (**aggregate**) to its default settings and set bit 0 (**active**) on dynamic aggregate partner port 1 in slot 7, enter:

```
-> no linkagg lacp port 7/1 partner admin-state active aggregate
```

Modifying the Partner Port Administrative Key

By default, the "administrative key" of the dynamic aggregate partner port is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp agg partner admin-key** command.

Configuring the Partner Port Administrative Key

You can configure the administrative key for the dynamic aggregate partner port to a value ranging from 0 to 65535 enter **linkagg lacp port**, the slot number, a slash (/), the port number, **partner admin-key**, and the user-specified partner port administrative key.

For example, to modify the administrative key of a dynamic aggregate group partner port 1 in slot 6 to 1000 enter:

```
-> linkagg lacp port 6/1 partner admin-key 1000
```

For example, to modify the administrative key of a dynamic aggregate group partner port 1 in slot 6, enter:

```
-> linkagg lacp port 6/1 partner admin-key 1000
```

Restoring the Partner Port Administrative Key

To remove a user-configured administrative key from the configuration set on a dynamic aggregate group partner port, use the **no** form of the **linkagg lacp agg partner admin-key** command by entering **no linkagg lacp agg**, the slot number, a slash (/), the port number, and **partner admin-key** keyword.

For example, to remove the user-configured administrative key from dynamic aggregate partner port 1 in slot 6, enter:

```
-> no linkagg lacp port 6/1 partner admin-key
```

Modifying the Partner Port System ID

By default, the partner port system ID (the MAC address used as the system ID on dynamic aggregate partner ports) is 00:00:00:00:00:00. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp port partner admin system-id** command.

Configuring the Partner Port System ID

You can configure the partner port system ID by entering **linkagg lacp port**, the slot number, a slash (/), the port number, **partner admin system-id**, and the user-specified partner administrative system ID (the MAC address in hexadecimal format).

For example, to modify the system ID of dynamic aggregate partner port 49 in slot 6 to **00:20:da:06:ba:d3**, enter:

```
-> linkagg lacp port 6/49 partner admin system-id 00:20:da:06:ba:d3
```

For example, to modify the system ID of dynamic aggregate partner port 49 in slot 6 to **00:20:da:06:ba:d3**, enter:

```
-> linkagg lacp port 6/49 partner admin system-id 00:20:da:06:ba:d3
```

Restoring the Partner Port System ID

To remove a user-configured system ID from a dynamic aggregate group partner port configuration use the **no** form of the **linkagg lacp port partner admin system-id** command by entering **linkagg lacp agg**, the slot number, a slash (/), the port number, and the **partner admin system-id** parameters.

For example, to remove a user-configured system ID from dynamic aggregate partner port 2 in slot 6, enter:

```
-> no linkagg lacp port 6/2 partner admin system-id
```

Modifying the Partner Port System Priority

By default, the administrative priority of a dynamic aggregate group partner port is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp agg partner system-priority** command.

Configuring the Partner Port System Priority

You can configure the administrative priority of a dynamic aggregate group partner port to a value ranging from 0 to 255 by entering **linkagg lacp port**, the slot number, a slash (/), the port number, **partner admin-system-priority**, and the user-specified administrative system priority.

For example, to modify the administrative priority of a dynamic aggregate partner port 49 in slot 4 to 100, enter:

```
-> linkagg lacp port 4/49 partner admin-system-priority 100
```

For example, to modify the administrative priority of dynamic aggregate partner port 49 in slot 4 to 100 and specify that the port is a Gigabit Ethernet port, enter:

```
-> linkagg lacp port 4/49 partner admin-system-priority 100
```

Restoring the Partner Port System Priority

To remove a user-configured system priority from a dynamic aggregate group partner port configuration use the **no** form of the **linkagg lacp agg partner system-priority** command by entering **lacp port**, the slot number, a slash (/), the port number, and **partner admin-system-priority**.

For example, to remove a user-configured system ID from dynamic aggregate partner port 3 in slot 4, enter:

```
-> no linkagg lacp port 4/3 partner admin-system-priority
```

Modifying the Partner Port Administrative Status

By default, the administrative status of a dynamic aggregate group partner port is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp port partner admin-port** command.

Configuring the Partner Port Administrative Status

You can configure the administrative status of a dynamic aggregate group partner port by entering **linkagg lacp port**, the slot number, a slash (/), the port number, **partner admin-port**, and the user-specified partner port administrative status.

For example, to modify the administrative status of dynamic aggregate partner port 1 in slot 7 to 200 you would enter:

```
-> linkagg lacp port 7/1 partner admin-port 200
```

For example, to modify the administrative status of dynamic aggregate partner port 1 in slot 7 to 200, enter:

```
-> linkagg lacp port 7/1 partner admin-port 200
```

Restoring the Partner Port Administrative Status

To remove a user-configured administrative status from a dynamic aggregate group partner port configuration use the **no** form of the **linkagg lacp port partner admin-port** command by entering **no linkagg lacp agg**, the slot number, a slash (/), the port number, and **partner admin-port**.

For example, to remove a user-configured administrative status from dynamic aggregate partner port 1 in slot 7 you would enter:

```
-> no linkagg lacp port 7/1 partner admin-port
```

Modifying the Partner Port Priority

The default partner port priority is 0. The following subsections describe how to configure a user-specified value and how to restore the value to its default value with the **linkagg lacp port partner admin port-priority** command.

Configuring the Partner Port Priority

To configure the partner port priority, enter **lacp agg**, the slot number, a slash (/), the port number, **partner admin-port priority**, and the user-specified partner port priority.

For example, to modify the port priority of dynamic aggregate partner port 3 in slot 4 to 100 you would enter:

```
-> linkagg lacp port 4/3 partner admin-port priority 100
```

For example, to modify the port priority of dynamic aggregate partner port 3 in slot 4 to 100, enter:

```
-> linkagg lacp port 4/3 partner admin-port priority 100
```

Restoring the Partner Port Priority

To remove a user-configured partner port priority from a dynamic aggregate group partner port configuration use the **no** form of the **linkagg lacp port partner admin port-priority** command by entering **no linkagg lacp port**, the slot number, a slash (/), the port number, **partner admin-port priority**.

For example, to remove a user-configured partner port priority from dynamic aggregate partner port 3 in slot 4 you would enter:

```
-> no linkagg lacp port 4/3 partner admin-port priority
```

Application Examples

Dynamic link aggregation groups are treated by the software on the switch as similar to individual physical ports. This section demonstrates the dynamic link aggregation feature by providing sample network configurations that use dynamic aggregation along with other software features. In addition, tutorials are provided that show how to configure these sample networks by using Command Line Interface (CLI) commands.

Sample Network Overview

The figure below shows two VLANs on Switch A that use two different link aggregation groups. VLAN 10 has been configured on dynamic aggregate group 5 with Spanning Tree Protocol (STP) with the highest priority (15) possible. And VLAN 12 has been configured on dynamic aggregate group 7 with 802.1Q tagging and 802.1p priority bit settings.

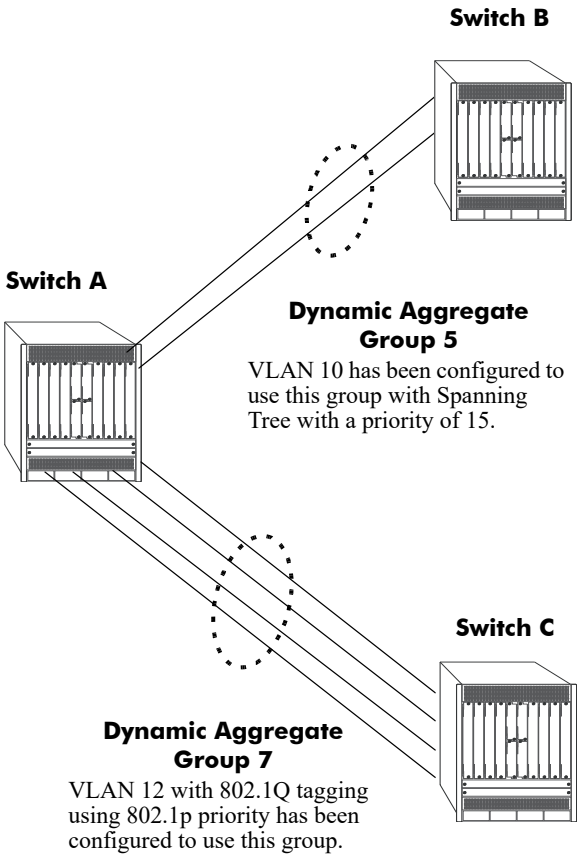


Figure 10-2 : Sample Network Using Dynamic Link Aggregation

The steps to configure VLAN 10 (Spanning Tree example) are described in "Link Aggregation and Spanning Tree Example" on page 10-26. The steps to configure VLAN 12 (802.1Q and 802.1p example) are described in "Link Aggregation and QoS Example" on page 10-27.

Note. Although you need to configure both the local (Switch A) and remote (Switches B and C) switches, only the steps to configure the local switch are provided since the steps to configure the remote switches are similar.

Link Aggregation and Spanning Tree Example

As shown in the figure on page 10-25, VLAN 10, which uses the Spanning Tree Protocol (STP) with a priority of 15, has been configured to use dynamic aggregate group 7. The actual physical links connect ports 3/9 and 3/10 on Switch A to ports 1/1 and 1/2 on Switch B. Follow the steps below to configure this network:

Note. Only the steps to configure the local (Switch A) are provided here since the steps to configure the remote (Switch B) are similar.

1 Configure dynamic aggregate group 5 by entering:

```
-> linkagg lacp agg 5 size 2
```

2 Configure ports 5/5 and 5/6 with the same actor administrative key (5) by entering:

```
-> linkagg lacp port 5/5-6 actor admin-key 5
```

3 Create VLAN 10 by entering:

```
-> vlan 10
```

4 If the Spanning Tree Protocol (STP) has been disabled on this VLAN (STP is enabled by default), enable it on VLAN 10 by entering:

```
-> vlan 10 stp enable
```

Note. *Optional*. Use the **show spantree ports** command to determine if the STP is enabled or disabled and to display other STP parameters. For example:

```
-> show spantree vlan 10 ports

Spanning Tree Port Summary for Vlan 10 AdmOper Man. Path Desig  FwPrim.AdmOp
Port  Pri St  St    mode Cost Cost   Role Tx  Port Cnx Cnx Desig Bridge ID
-----+---+---+----+----+-----+-----+----+---+-----+---+---+--------------------
3/13   7  ENA FORW No   100   0     DESG 1    3/13EDG  NPT 000A-00:d0:95:6b:0a:c0
2/10   7  ENA FORW No   19    0     DESG 1    2/10PTP  PTP 000A-00:d0:95:6b:0a:c0
5/2    7  ENA  DIS No   0     0     DIS  0    5/2 EDG  NPT 0000-00:00:00:00:00:00
0/5    7  ENA FORW No   4     0     DESG 1    0/10PTP  PTP 000A-00:d0:95:6b:0a:c0
```

In the example above the link aggregation group is indicated by the “0” for the slot number.

5 Configure VLAN 10 (which uses dynamic aggregate group 5) to the highest (15) priority possible by entering:

```
-> spantree vlan 10 linkagg 5 priority 15
```

6 Repeat steps 1 through 5 on Switch B. Substitute the port numbers of the commands with the appropriate port numbers of Switch B.

Link Aggregation and QoS Example

As shown in the figure on page 10-25, VLAN 12, which uses 802.1Q frame tagging and 802.1p prioritization, has been configured to use dynamic aggregate group 7. The actual physical links connect ports 4/1, 4/2, 4/3, and 4/4 on Switch A to ports 1/1, 1/2, 1/3, and 1/4 on Switch C. Follow the steps below to configure this network:

Note. Only the steps to configure the local (Switch A) switch are provided here since the steps to configure the remote (Switch C) switch would not be significantly different.

1 Configure dynamic aggregate group 7 by entering:

```
-> linkagg lacp agg 7 size 4
```

2 Configure ports 4/1, 4/2, 4/3, and 4/4 the same actor administrative key (7) by entering:

```
-> lacp agg 4/1-4 actor admin-key 7
```

3 Create VLAN 12 by entering:

```
-> vlan 12
```

4 Configure 802.1Q tagging with a tagging ID (VLAN ID) of 12 on dynamic aggregate group 7 by entering:

```
-> vlan 12 members 7
```

5 If the QoS Manager has been disabled (it is enabled by default) enable it by entering:

```
-> qos enable
```

Note. *Optional*. Use the **show qos config** command to determine if the QoS Manager is enabled or disabled.

6 Configure a policy condition for VLAN 12 called “vlan12_condition” by entering:

```
-> policy condition vlan12_condition destination vlan 12
```

7 Configure an 802.1p policy action with the highest priority possible (7) for VLAN 12 called “vlan12_action” by entering:

```
-> policy action vlan12_action 802.1P 7
```

8 Configure a QoS rule called “vlan12_rule” by using the policy condition and policy rules you configured in steps **8** and **9** above by entering:

```
-> policy rule vlan12_rule enable condition vlan12_condition action
vlan12_action
```

9 Enable your 802.1p QoS settings by entering **qos apply** as shown below:

```
-> qos apply
```

10 Repeat steps 1 through 9 on Switch C. Use the same commands as mentioned in the previous steps. Substitute the port numbers of the commands with the appropriate port numbers of Switch C.

Note. If you do not use the **qos apply** command any QoS policies previously configured, are lost on the next switch reboot.

Displaying Dynamic Link Aggregation Configuration and Statistics

You can use Command Line Interface (CLI) **show** commands to display the current configuration and statistics of link aggregation. These commands include the following:

show linkagg	Displays information on link aggregation groups.
show linkagg port	Displays information on link aggregation ports.

When you use the **show linkagg** command without specifying the link aggregation group number and when you use the **show linkagg port** command without specifying the slot and port number, these commands provide a "global" view of switch-wide link aggregate group and link aggregate port information, respectively.

For example, to display global statistics on all link aggregate groups (both dynamic and static), enter:

```
-> show linkagg agg
```

A screen similar to the following would be displayed:

```
Number  Aggregate  SNMP Id  Size  Admin State  Oper State     Att/Sel Ports
-------+----------+--------+----+-------------+-------------+-------------
   1     Static    40000001   8     ENABLED      UP            2   2
   2     Dynamic   40000002   4     ENABLED      DOWN          0   0
   3     Dynamic   40000003   8     ENABLED      DOWN          0   2
   4     Static    40000005   2     DISABLED     DOWN          0   0
```

When you use the **show linkagg** command with the **agg** keyword and the link aggregation group number and when you use the **show linkagg port** command with the slot and port number, these commands provide detailed views of the link aggregate group and port information, respectively. These detailed views provide excellent tools for diagnosing and troubleshooting problems.

For example, to display detailed statistics for port 1 in slot 2 that is attached to dynamic link aggregate group 1, enter:

```
-> show linkagg port 2/1
```

A screen similar to the following would be displayed:

```
Dynamic Aggregable Port
   SNMP Id                        : 2001,
   Slot/Port                      : 2/1,
   Administrative State           : ENABLED,
   Operational State              : DOWN,
   Port State                     : CONFIGURED,
   Link State                     : DOWN,
   Selected Agg Number            : NONE,
   Primary port                   : UNKNOWN,
LACP
   Actor System Priority          : 10,
   Actor System Id                : [00:d0:95:6a:78:3a],
   Actor Admin Key                : 8,
   Actor Oper Key                 : 8,
   Partner Admin System Priority  : 20,
   Partner Oper System Priority   : 20,
   Partner Admin System Id        : [00:00:00:00:00:00],
   Partner Oper System Id         : [00:00:00:00:00:00],
```

```
Partner Admin Key               : 8,
Partner Oper Key                : 0,
Attached Agg Id                 : 0,
Actor Port                      : 7,
Actor Port Priority             : 15,
Partner Admin Port              : 0,
Partner Oper Port               : 0,
Partner Admin Port Priority     : 0,
Partner Oper Port Priority      : 0,
Actor Admin State               : act1.tim1.agg1.syn0.col0.dis0.def1.exp0,
Actor Oper State                : act1.tim1.agg1.syn0.col0.dis0.def1.exp0,
Partner Admin State             : act0.tim0.agg1.syn1.col1.dis1.def1.exp0,
Partner Oper State              : act0.tim0.agg1.syn0.col1.dis1.def1.exp0
```

See the “Link Aggregation Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for complete documentation of **show** commands for link aggregation.

11 Configuring Dual-Home Links

Dual-Home Link (DHL) is a high availability feature that provides fast failover between core and edge switches without implementing Spanning Tree. The OmniSwitch provides the following method for implementing a DHL solution:

DHL Active-Active—an edge technology that splits a number of VLANs between two active links. The forwarding status of each VLAN is modified by DHL to prevent network loops and maintain connectivity to the core when one of the links fails. This solution does not require link aggregation.

In This Chapter

This chapter describes the basic components of DHL solutions and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Information and procedures described in this chapter include:

- "Dual-Home Link Active-Active Defaults" on page 11-2
- "Dual-Home Link Active-Active" on page 11-3.
- "Configuring DHL Active-Active" on page 11-6.
- "Dual-Home Link Active-Active Example" on page 11-8.
- "Displaying the Dual-Home Link Configuration" on page 11-12

Dual-Home Link Active-Active Defaults

The table below lists default values for dual-home link aggregate groups.

Parameter Description	Command	Default Value/Comments
DHL session ID	**dhl name**	If a name is not assigned to a DHL session, the session is configured as DHL-1
Admin state of DHL session	**dhl admin-state**	disable
Configure a port/link agg as DHL	**dhl linka linkb**	NA
Configure a VLAN-MAP	**dhl vlan-map linkb**	NA
Pre-emption timer for the DHL session	**dhl pre-emption-time**	30 seconds

Dual-Home Link Active-Active

Dual-Home Link (DHL) Active-Active is a high availability feature that provides fast failover between core and edge switches without using Spanning Tree. To provide this functionality, DHL Active-Active splits a number of VLANs between two active links. The forwarding status of each VLAN is modified by DHL to prevent network loops and maintain connectivity to the core when one of the links fails.

The DHL Active-Active feature, however, is configurable on regular switch ports and on logical link aggregate ports (linkagg ID) instead of just LACP aggregated ports. In addition, the two DHL links are both active, as opposed to the active and standby mode used with LACP.

DHL Active-Active Operation

A DHL Active-Active configuration consists of the following components:

- A DHL session. Only one session per switch is allowed.
- Two DHL links associated with the session (link A and link B). A physical switch port or a logical link aggregate (linkagg) ID are configurable as a DHL link.
- A group of VLANs (or pool of common VLANs) in which each VLAN is associated (802.1q tagged) with both link A and link B.
- A VLAN-to-link mapping that specifies which of the common VLANs each DHL link services. This mapping prevents network loops by designating only one active link for each VLAN, even though both links remain active and are associated with each of the common VLANs.

When one of the two active DHL links fails or is brought down, the VLANs mapped to that link are then forwarded on the remaining active link to maintain connectivity to the core. When the failed link comes back up, DHL waits a configurable amount of time before the link resumes forwarding of its assigned VLAN traffic.

The following diagram shows how DHL works when operating in a normal state (both links up) and when operating in a failed state (one link is down):

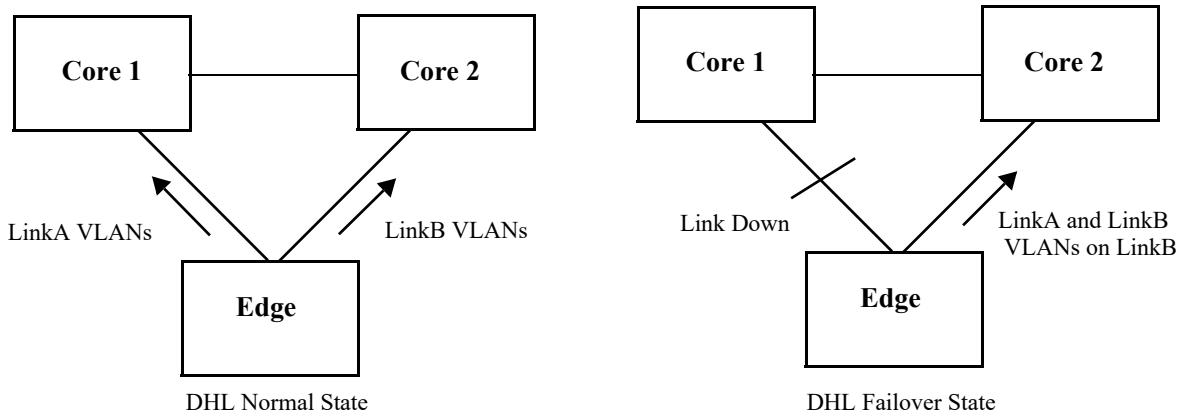


Figure 11-1 : DHL Active-Active Operation

Protected VLANs

A protected VLAN is one that is assigned to both links in a DHL session. This means that if the link to which the VLAN is mapped fails, the VLAN is moved to the other active DHL link to maintain connectivity with the core switches.

Any VLAN that is only assigned to one of the DHL links is considered an unprotected VLAN. This type of VLAN is not eligible for DHL support if the link to which the VLAN is assigned fails.

DHL Port Types

DHL is supported on the following port types:

- Physical switch ports.
- Logical link aggregate ports (linkagg ID).
- LPS ports
- NNI ports
- IPM VLAN ports
- DHCP Snooping ports
- IP Source filtering ports.

DHL is not supported on the following port types:

- Any port that is a member of a link aggregate.
- Mobile ports
- 802.1x ports
- GVRP ports.
- UNI ports
- Ports that are enabled for transparent bridging.

Note. No CLI error message is displayed when DHL is configured using a port type that is not supported.

DHL Pre-Emption Timer

The DHL pre-emption timer specifies the amount of time to wait before a failed link that has recovered can resume servicing VLANs that are mapped to that link. This time value is configured on a per-DHL session basis.

MAC Address Flushing

Spanning Tree flushes the MAC address table when a topology change occurs that also changes the forwarding topology. The MAC addresses are then relearned according to the new forwarding topology. This prevents MAC address entries from becoming stale (entries contain old forwarding information).

When a port is configured as a DHL Active-Active link, Spanning Tree is automatically disabled on the port. Since Spanning Tree is not used, a changeover from one DHL link to the other does not trigger a

topology change event and the MAC address table is not automatically flushed. This can create stale MAC address entries that are looking for end devices over the wrong link.

To avoid stale MAC address entries in the forwarding tables of the core switches, some type of communication needs to occur between the edge uplink switch and the core switches. The DHL Active-Active feature provides two methods for clearing stale MAC address entries: MVRP Enhanced Operation or Raw Flooding. Selecting which one of these methods to use is done on a per-DHL session basis.

MVRP Enhanced Operation

The switch uses an enhanced Multiple VLAN Registration Protocol (MVRP) operation to refresh core MAC address tables as follows:

- For each uplink port, the switch issues joins for each VLAN that is active on that port. This causes the core switch to only register those VLANs that are active on each link based on the DHL configuration.
- When one of the DHL links fails, the other link issues joins to establish connectivity for the VLANs that were serviced by the failed link. These new joins contain the “new” flag set, which are forwarded by the core devices and trigger a flush of the MAC addresses on the core network for the joined VLANs.
- When a failed DHL link recovers, the link issues new joins to re-establish connectivity for the VLANs the link was servicing before the link went down. These new joins also trigger a flush of the MAC addresses on the core network for the joined VLANs.

The switch interacts normally with the core and other devices for MVRP, treating the DHL VLANs on each uplink port as a fixed registration. This approach requires core devices that support MVRP.

Raw Flooding

When a DHL link fails or recovers and Raw Flooding is enabled for the DHL session, the switch performs the following tasks to trigger MAC movement:

- Identify a list of MAC addresses within the effected VLANs that were learned on non-DHL ports (MAC addresses that were reachable through the effected VLANs).
- Create a tagged packet for each of these addresses. The SA for the packet is one of the MAC addresses from the previously-generated list; the VLAN tag is the resident VLAN for the MAC address; the DA is set for broadcast (all Fs); the body is just filler.
- Transmit the generated packet once for each VLAN-MAC address combination. These packets are sent on the link that takes over for the failed link or on a link that has recovered from a failure.

The MAC movement triggered by the Raw Flooding method clears any stale MAC entries. However, flooded packets are often assigned a low priority and the switch may filter such packets in a highly utilized network.

DHL Configuration Guidelines

Review the following guidelines before attempting to configure a DHL setup:

- Make sure that DHL linkA *and* linkB are associated with each VLAN protected by the DHL session. Any VLAN not associated with either link or only associated with one of the links is unprotected.
- DHL linkA *and* linkB must belong to the same default VLAN. In addition, select a default VLAN that is one of the VLANs protected by the DHL session. For example, if the session is going to protect VLANs 10-20, then assign one of those VLANs as the default VLAN for linkA and linkB.
- Only one DHL session per switch is allowed. Each session can have only two links (linkA and linkB). Specify a physical switch port or a link aggregate (linkagg) ID as a DHL link. The same port or link aggregate is not configurable as both linkA or linkB.
- The administrative state of a DHL session is not configurable until a linkA port and a linkB port are associated with the specified DHL session ID number.
- Spanning Tree is automatically disabled on each link when the DHL session is enabled.
- Do not change the link assignments for the DHL session while the session is enabled.
- Configuring a MAC address flush method (MVRP or Raw Flooding) is recommended if the DHL session ports span across switch modules. This configuration improves convergence time.
- Enabling the registrar mode as “forbidden” is recommended before MVRP is enabled on DHL links.

Configuring DHL Active-Active

Configuring a DHL Active-Active setup requires the following tasks.

1 Configure a set of VLANs that the two DHL session links service.

```
-> vlan 100-110
```

2 Identify two ports or link aggregates that serve as the links for the DHL session then assign both links to the same default VLAN. Make sure the default VLAN is one of the VLANs created in Step 1. For example, the following commands assign VLAN 100 as the default VLAN for port 1/1/10 and linkagg 5:

```
-> vlan 100 members port 1/1/10 untagged
-> vlan 100 members linkagg 5 untagged
```

3 Associate (802.1q tag) the ports identified in Step 2 to each one of the VLANs created in Step 1, except for the default VLAN already associated with each port. For example, the following commands associate port 1/1/10 and linkagg 5 with VLANs 101-110:

```
-> vlan 101-110 members port 1/1/10 tagged
-> vlan 101-110 members linkagg 5 tagged
```

In the above command example, port 1/1/10 and linkagg 5 are only tagged with VLANs 101-110 because VLAN 100 is already the default VLAN for both ports.

4 Create a DHL session using the **dhl name** command. For example:

```
-> dhl 10
```

5 Configure the pre-emption (recovery) timer for the DHL session using the **dhl pre-emption-time** command. By default, the timer is set to 30 seconds, so it is only necessary to change this parameter if the default value is not sufficient. For example, the following command changes the timer value 500 seconds:

```
-> dhl 10 pre-emption-time 500
```

6 Configure the MAC address flushing method for the DHL session using the **dhl mac-flushing** command and specify either the **raw** or **mvrp** parameter option. By default, the MAC flushing method is set to none. For example, the following command selects the MVRP method:

```
-> dhl 10 mac-flushing mvrp
```

7 Configure two links (linkA and linkB) for the DHL session using the **dhl linka linkb** command. Specify the ports identified in Step 1 as linkA and linkB. For example:

```
-> dhl 10 linka linkagg 5 linkb port 1/1/10
```

8 Select VLANs from the set of VLANs created in Step 2 and map those VLANs to linkB using the **dhl vlan-map linkb** command. Any VLAN not mapped to linkB is automatically mapped to linkA. By default, all VLANs are mapped to linkA. For example, the following command maps VLANs 11-20 to linkB:

```
-> dhl 10 vlan-map linkb 11-20
```

9 Administratively enable the DHL session using the **dhl admin-state** command. For example:

```
-> dhl 10 admin-state enable
```

See "Dual-Home Link Active-Active Example" on page 11-8 for a DHL application example.

Dual-Home Link Active-Active Example

The figure below shows two ports (1/1/10 and 1/1/12) that serve as link A and link B for a DHL session configured on the Edge switch. Both ports are associated with VLANs 1-10, where VLAN 1 is the default VLAN for both ports. The odd numbered VLANs (1, 3, 5, 7, 9) are mapped to link A and the even numbered VLANs (2, 4, 6, 8, 10) are mapped to link B. Spanning Tree is disabled on both ports.

Both DHL links are active and provide connectivity to the Core switches for the VLANs to which each link is mapped. If one link fails or is brought down, the VLANs mapped to the failed link are switched over to the remaining active link to maintain connectivity for those VLANs. For example, if link A goes down, VLANs 1, 3, 5, 7, and 9 are switch over and carried on link B.

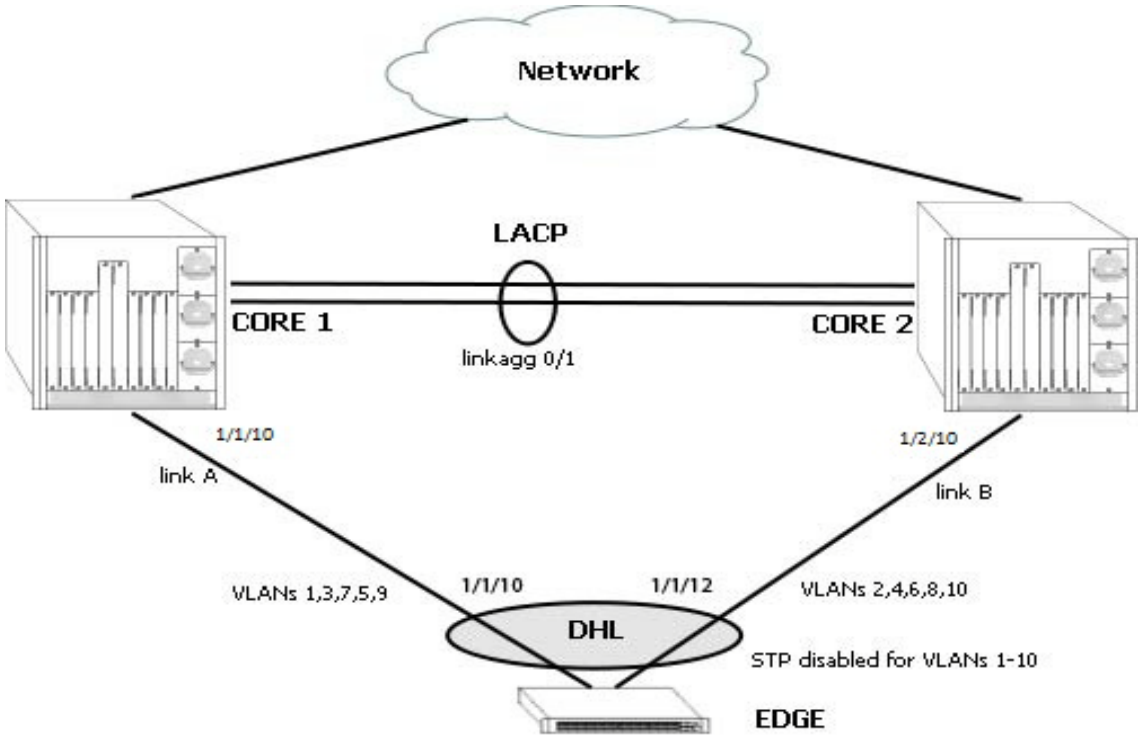


Figure 11-2 : Dual-Home Link Active-Active Example

Follow the steps below to configure this example DHL configuration.

Edge Switch:

1 Create VLANs 2-10.

```
-> vlan 2-10
```

2 Configure 802.1q tagging on VLANs 2-10 for port 1/1/10. Because VLAN 1 is the default VLAN for port 1/1/10, there is no need to tag VLAN 1.

```
-> vlan 2-10 members port 1/1/10 tagged
```

3 Configure 802.1q tagging on the VLANs 2-10 for port 1/1/12. Because VLAN 1 is the default VLAN for port 1/1/12, there is no need to tag VLAN 1.

```
-> vlan 2-10 members port 1/1/12 tagged
```

4 Configure a session ID and an optional name for the DHL session.

```
-> dhl 1 name dhl_session1
```

5 Configure port 1/1/10 and port 1/1/12 as the dual-home links (linkA, linkB) for the DHL session.

```
-> dhl 1 linkA port 1/1/10 linkB port 1/1/12
```

6 Map VLANs 2, 4, 6, 8, and 10 to DHL linkB.

```
-> dhl 1 vlan-map linkb 2
-> dhl 1 vlan-map linkb 4
-> dhl 1 vlan-map linkb 6
-> dhl 1 vlan-map linkb 8
-> dhl 1 vlan-map linkb 10
```

7 Specify Raw Flooding as the MAC flushing technique to use for this DHL session.

```
-> dhl 1 mac-flushing raw
```

8 Enable the administrative state of the DHL session using the following command:

```
-> dhl 1 admin-state enable
```

Core Switches:

1 Create VLANs 2-10.

```
-> vlan 2-10
```

2 Configure 802.1q tagging on VLANs 2-10 for port 1/10 on the Core 1 switch. VLAN 1 is the default VLAN for port 1/10, so there is no need to tag VLAN 1.

```
-> vlan 2-10 members port 1/1/10 tagged
```

CLI Command Sequence Example

The following is an example of what the example DHL configuration commands look like entered sequentially on the command line:

Edge Switch:

```
-> vlan 2-10
-> vlan 2-10 members port 1/1/10 tagged
-> vlan 2-10 members port 1/1/12 tagged
-> dhl 1 name dhl_session1
-> dhl 1 linkA port 1/1/10 linkB port 1/1/12
-> dhl 1 vlan-map linkb 2
-> dhl 1 vlan-map linkb 4
-> dhl 1 vlan-map linkb 6
-> dhl 1 vlan-map linkb 8
-> dhl 1 vlan-map linkb 10
-> dhl 1 mac-flushing raw
-> dhl 1 admin-state enable
```

Core 1 Switch:

```
-> vlan 2-10
```

Core 2 Switch:

```
-> vlan 2-10
```

Recommended DHL Active-Active Topology

The following is an example of a recommended topology for Dual-Home Link Active-Active.

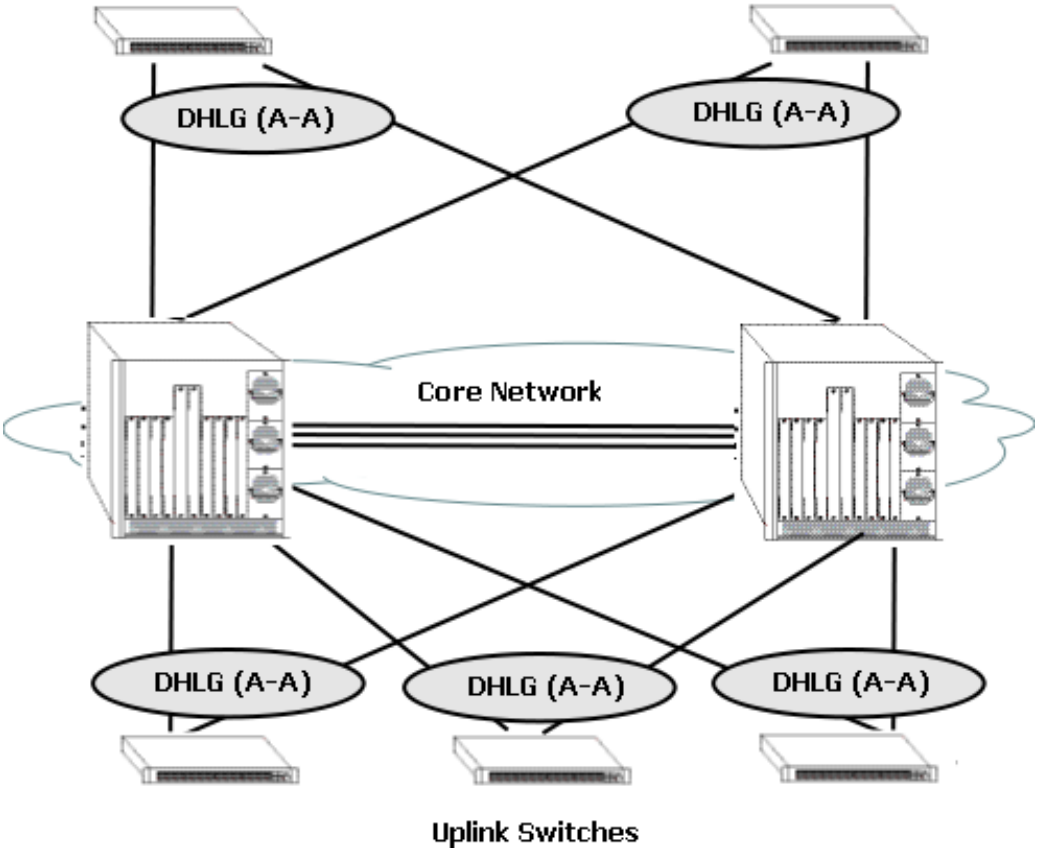


Figure 11-3 : Recommended DHL Active-Active Topology

In the above topology, all uplinked switches are connected to the core network through redundant links, and the links are configured to use DHL Active-Active. Spanning Tree is disabled on all the DHL enabled ports of the uplinked devices.

Unsupported DHL Active-Active Topology (Network Loops)

The following is an example of an unsupported topology for Dual-Homed Link Active-Active.

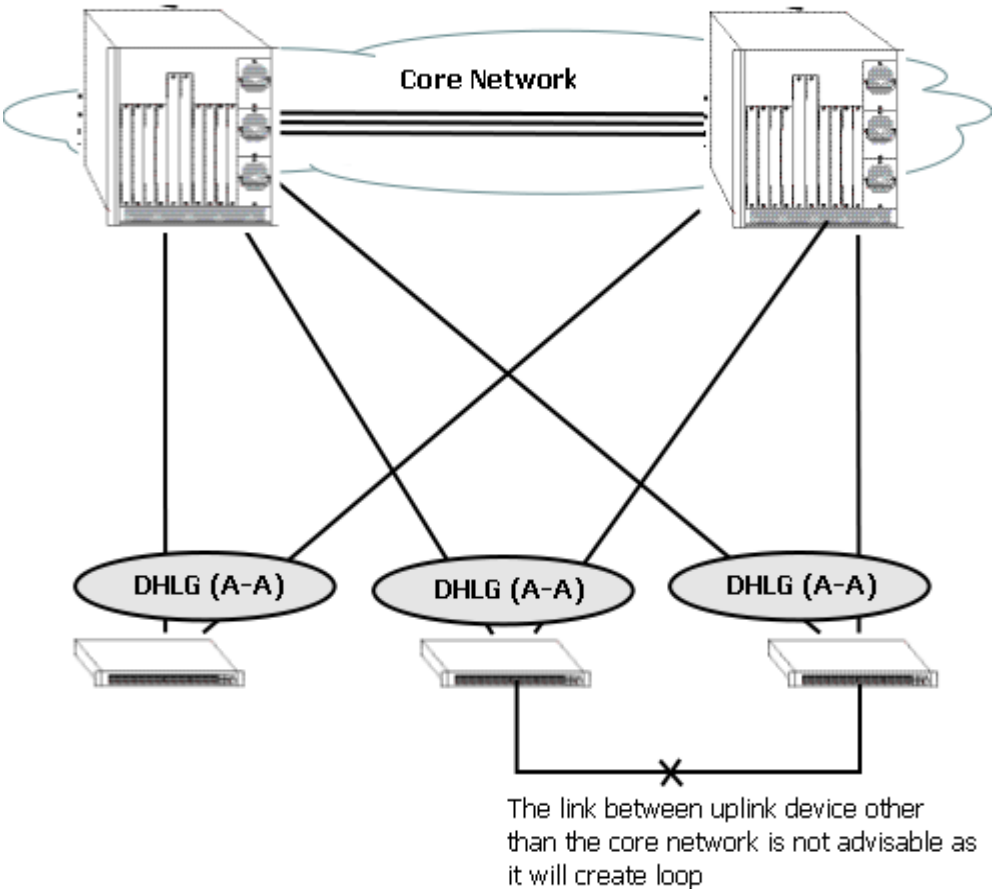


Figure 11-4 : Unsupported DHL Active-Active Topology (Network Loops)

In the above topology, the link between the uplink device other than core network is not recommended as it creates a loop in the network. This topology violates the principle that uplink switches can only be connected to the network cloud through the core network.

Displaying the Dual-Home Link Configuration

You can use Command Line Interface (CLI) **show** commands to display the current configuration and statistics of link aggregation. These commands include the following:

show linkagg	Displays information on link aggregation groups.
show linkagg port	Displays information on link aggregation ports.
show dhl	Displays the global status of the DHL configuration.
show dhl	Displays information about a specific DHL session.
show dhl link	Displays information about a specific DHL link, for example linkA or linkB and the VLAN details of the specified link.

Note. See the "Link Aggregation Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for complete documentation of **show** commands for link aggregation.

12 Configuring ERP

The ITU-T G.8032/Y.1344 Ethernet Ring Protection (ERP) switching mechanism is a self-configuring algorithm that maintains a loop-free topology while providing data path redundancy and network scalability. ERP provides fast recovery times for Ethernet ring topologies by utilizing traditional Ethernet MAC and bridge functions.

Loop prevention is achieved by allowing traffic to flow on all except one of the links within the protected Ethernet ring. This link is blocked and is referred to as the Ring Protection Link (RPL). When a ring failure condition occurs, the RPL is unblocked to allow the flow of traffic to continue through the ring.

The OmniSwitch also supports ERPv2 according to the ITU-T recommendation G.8032 03/2010. ERPv2 implementation helps maintain a loop-free topology in multi-ring and ladder networks that contain interconnection nodes, interconnected shared links, master rings and sub-rings.

The following chapter details the different functionalities and configuration settings required for ERP and ERPv2.

In This Chapter

This chapter provides an overview about how Ethernet Ring Protection (ERP) works and how to configure its parameters through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following information and configuration procedures are included in this chapter:

- "ERP Overview" on page 12-3.
- "Interaction With Other Features" on page 12-9
- "Quick Steps for Configuring ERP with Standard VLANs" on page 12-10.
- "Quick Steps for Configuring ERP with VLAN Stacking" on page 12-11
- "ERP Configuration Overview and Guidelines" on page 12-12
- "ERPv2 Configuration Overview and Guidelines" on page 12-17.
- "Sample Ethernet Ring Protection Configuration" on page 12-21.
- "Sample ERPv2 Ring Configuration" on page 12-23.
- "Configuring ERP over SPB" on page 12-26
- "Verifying the ERP Configuration" on page 12-30

ERP Defaults

ERP default settings:

Parameter Description	Command	Default
ERP ring status	**erp-ring**	Disabled
RPL status for the node	**erp-ring rpl-node**	Disabled
The wait-to-restore timer value for the RPL node	**erp-ring wait-to-restore**	5 minutes
The guard-timer value for the ring node	**erp-ring guard-timer**	50 centi-seconds
The NNI-SVLAN association type	**ethernet-service svlan nni**	STP

ERPv2 default settings:

The Ethernet Ring Protection (ERP) Ring Virtual Channel.	**erp-ring virtual-channel**	Enabled
Revertive mode on a specified node.	**erp-ring revertive**	Enabled

ERP Overview

Ethernet Ring Protection (ERP) is a protection switching mechanism for Ethernet ring topologies, such as multi-ring and ladder networks. This implementation of ERP is based on the Recommendation ITU-T G.8032/Y.1344 and uses the ring Automatic Protection Switching (APS) protocol to coordinate the prevention of network loops within a bridged Ethernet ring.

Loop prevention is achieved by allowing the traffic to flow on all but one of the links within the protected Ethernet ring. This link is blocked and is referred to as the Ring Protection Link (RPL). When a ring failure condition occurs, the RPL is unblocked to allow the flow of traffic to continue through the ring.

One designated node within the ring serves as the RPL owner and is responsible for blocking the traffic over the RPL. When a ring failure condition occurs, the RPL owner is responsible for unblocking the RPL so that the link can forward traffic to maintain ring connectivity.

The ERPv2 capability supports multi-ring and ladder networks with interconnection nodes, interconnected shared links, master rings and sub-rings. The following features are also supported:

- R-APS Virtual Channel
- Revertive/Non-Revertive modes

A shared link can be a part of one master ring. The sub-rings connected to the interconnection nodes are open. The sub-rings cannot use shared links.

ERP and ERPv2 Terms

Ring Protection Link (RPL) and **RB**—A designated link between two ring nodes that is blocked to prevent a loop on the ring. RB specifies a blocked RPL.

RPL Owner—A node connected to an RPL. This node blocks traffic on the RPL during normal ring operations and activates the link to forward traffic when a failure condition occurs on another link in the ring.

RMEPID—Remote Maintenance End Point identifier.

Link Monitoring—Ring links are monitored using standard ETH (Ethernet Layer Network) CC OAM messages (CFM). Note that for improved convergence times, this implementation also uses Ethernet link up and link down events.

Signal Fail (SF)—Signal Fail is declared when a failed link or node is detected.

No Request (NR)—No Request is declared when there are no outstanding conditions (for example, SF) on the node.

Ring APS (Automatic Protection Switching) Messages—Protocol messages defined in Y.1731 and G.8032 that determine the status of the ring.

ERP Service VLAN—Ring-wide VLAN used exclusively for transmission of messages, including R-APS messages for Ethernet Ring Protection.

ERP Protected VLAN—A VLAN that is added to the ERP ring. ERP determines the forwarding state of protected VLANs.

FDB—The Filtering Database that stores filtered data according to the R-APS messages recieved. This database also maintains an association table that identifies the master rings for a given sub-ring.

BPR—The Blocked Port Reference that identifies the ring port (0 for interconnection node or sub-ring, 1 for master ring) that is blocked. The BPR status is used in all R-APS messages.

CCM—When an Ethernet ring contains no ERP capable nodes, CCM (Continuity Check Messages) are required to monitor the ring-port connectivity across the L2 network.

MEG and **MEL**—The switches in the Management Entity Group with given priority as MEG level (MEL).

NR and **SF**—Not Reachable and Signal Failure specify the status messages that can be sent as part of the R-APS messages.

ERP Timers

Wait To Restore (WTR) Timer. This timer is used by the RPL to verify stability of the Ethernet ring. WTR Timer determines the number of minutes the RPL switch waits before returning the RPL ports to a blocked state after the ring has recovered from a link failure.

Some important points about the WTR Timer are as follows:

- The timer is started when the RPL node receives an R-APS (NR) message that indicates ring protection is no longer required.
- The timer is stopped when the RPL owner receives an R-APS (SF) message while WTR is running, which indicates that an error still exists in the ring.
- When the time runs out, the RPL port is blocked and an R-APS (NR, RB) message is transmitted from both the ring ports to indicate that the RPL is blocked.
- Refer to the “Ethernet Ring Protection Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for timer defaults and valid ranges.

Guard Timer. When the failed link recovers, a ring node starts the Guard Timer. The Guard Timer is used to prevent the ring nodes from receiving outdated R-APS messages that are no longer relevant.

Some important points about the Guard Timer are as follows:

- When the Guard Timer is running, any R-APS messages received are not forwarded.
- The Guard Timer value must be greater than the maximum expected forwarding delay time for which it takes one R-APS message to circulate around the ring. This calculated value is required to prevent any looping scenarios within the ring.
- Refer to the “Ethernet Ring Protection Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for timer defaults and valid ranges.

ERP Basic Operation

ERP operates over standard Ethernet interfaces that are physically connected in a ring topology. It uses an Automatic Protection Switching (APS) protocol to coordinate protection and recovery switching mechanisms over the Ethernet ring.

In an Ethernet ring, each node is connected to two adjacent nodes using two independent links called ring links. A ring link is bound by two adjacent nodes on ports called ring ports. The ring nodes support standard FDB (Filtering database) MAC learning, forwarding, flush behavior, and port blocking and unblocking mechanisms.

The Ethernet ring has a designated Ring Protection Link (RPL), which is blocked under normal conditions in order to avoid forming a loop in the ring. When a link or port failure is detected, a Signal Failure (SF) message is sent on the ring to inform other ring nodes of the failure condition. At this point the ring is operating in protection mode. When this mode is invoked, the RPL is unblocked forming a new traffic pattern on the ring, (for example, traffic is accommodated on the RPL but blocked on the failed link). The node responsible for blocking and unblocking the RPL is called the RPL Owner.

ERP Ring Modes

A ring operates in one of two modes: idle (normal operation; all links up and RPL is blocked) and protection (protection switching activated; a ring failure has triggered the RPL into a forwarding state).

The following illustration shows an example of an ERP ring operating in the idle mode; all ring nodes are up and the RPL is blocked:

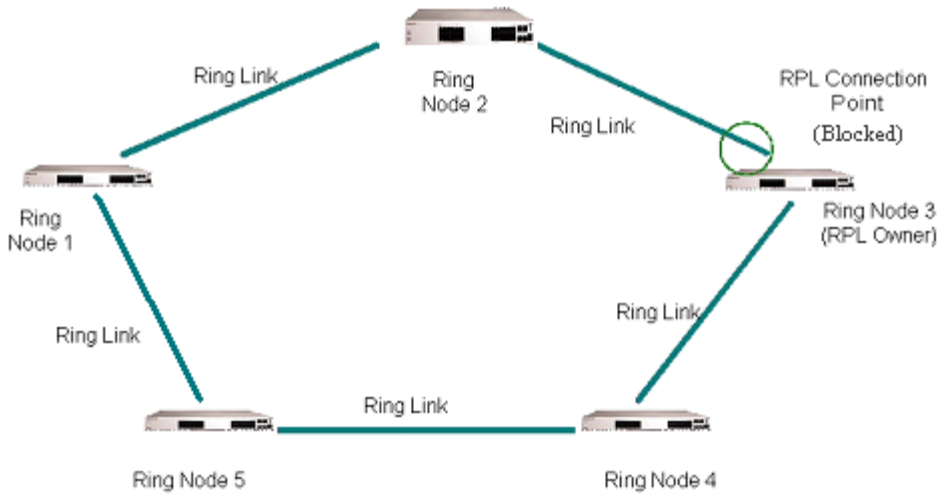


Figure 12-1 : Normal Mode

If a link or node failure occurs in the ring shown in the above illustration, the ring transitions as follows into the protection mode:

- Nodes adjacent to the failure detect and report the failure using the R-APS (SF) message.
- The R-APS (SF) message triggers the RPL owner to unblock the RPL.
- All nodes in the ring flush all the dynamic MAC addresses learned on their ring ports.

The ring is now operating in the protection mode, as shown below:

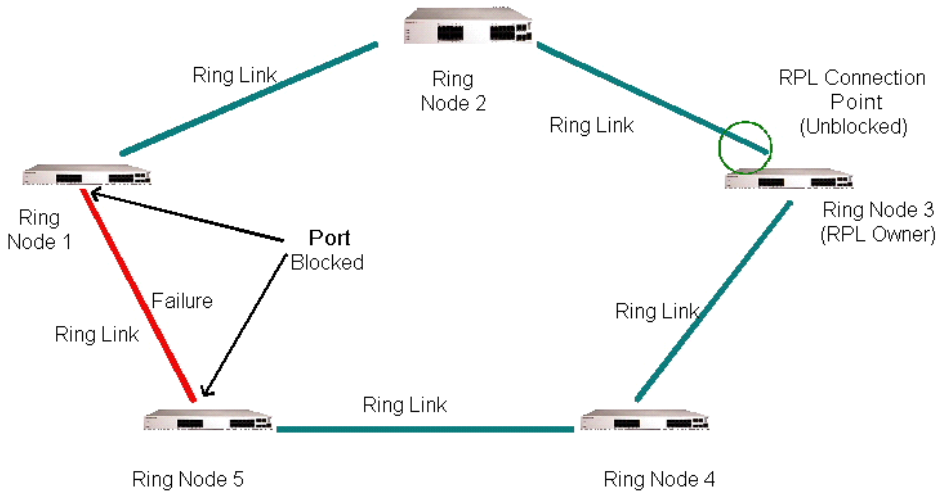


Figure 12-2 :Protection Mode

When the failed link shown in the above illustration recovers, the ring transitions as follows back to the idle mode:

- Nodes adjacent to the recovered link initiate an R-APS (NR) message and start the Guard Timer.
- When the RPL owner receives the R-APS (NR) message, it starts the Wait-To-Restore timer (WTR), which is the set period of time that must elapse before the RPL owner blocks the RPL.
- Once the WTR timer expires, the RPL owner blocks the RPL and transmits an R-APS (NR, RB) message indicating that RPL is blocked (RB).
- On receiving the R-APS (NR, RB) message, ring nodes flush all the dynamic MAC addresses learned on their ring ports and unblock any previously blocked ports.
- The ring is now operating in the idle mode. The RPL is blocked and all other ring links are operational.

Overlapping Protected VLANs Between ERP Rings on same Node

In a network where all connected nodes cannot belong to a single ERP ring, the OmniSwitch supports multiple ERP rings with a single shared node. The network example below shows two ERP rings connected with a shared node.

ERPv2 Basic Operation

The enhanced ERPv2 functionality supports multi-ring and ladder networks that contain interconnection nodes, interconnected shared links, master rings and sub-rings. Multiple sub-tending rings are supported over the same physical ring.

A shared link can only be part of the master ring. The sub-rings connected to the interconnection nodes are not closed and cannot use the shared links.

Consider the following OmniSwitch multi-ring and ladder network with the Master or Major Ring with five ring nodes. The Sub-ring, ladder networks, RPLs and Shared Links are also depicted as part of the illustration.

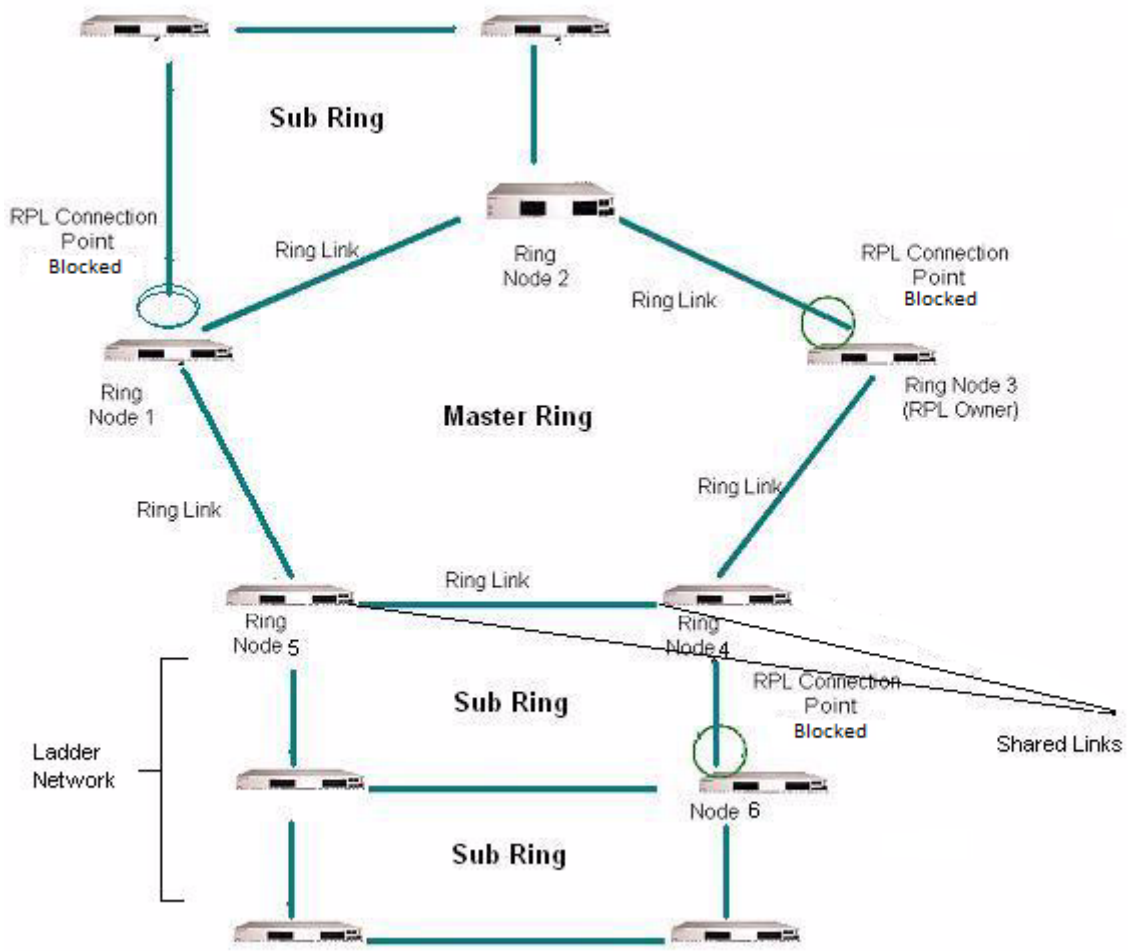


Figure 12-3 :ERPv2 on Multi Ring and Ladder Network with RPLs and Shared Links

R-APS Virtual Channel

ERPv2 supports two implementation options for R-APS control channel of the sub-ring.

- **Virtual Channel Enabled** - R-APS messages are encapsulated and transmitted over an R-APS Virtual channel configured on the major ring.
- **Virtual Channel Disabled** - R-APS messages are terminated at the interconnection nodes but not blocked at RPL of the sub-ring. RPL ports are unblocked when all nodes are active (there is no failed node).

For details on how to enable and disable R-APS virtual channel, see the section - "Enabling and Disabling R-APS Virtual Channel" on page 12-19

The R-APS channels are not shared across rings. Each ring must have its own R-APS Channel.

- The R-APS virtual channels of the sub rings are automatically **closed** using the master ring. R-APS messages from the sub ring on the interconnection node are forwarded as normal data to and only to the master ring ports.
- The R-APS messages use a static destination mac-address of 01-19-A7-00-00-00. R-APS messages must be tagged in order to identify the ring ID.

Note. The Service VLAN must be tagged, no support of "untagged" service VLAN. The sub ring and master ring cannot use the same service VLAN.

Revertive / Non-Revertive Mode

Revertive mode is configured for compatibility between ERPv1 and ERPv2 nodes in the same ring. When the ERPv2 node is operating with ERP v1 node in the same ring, it operates in revertive mode regardless of user configuration.

Non-Revertive mode: Under non-revertive mode, when the failure condition recovers, the port that has been blocked stays blocked and the unblocked RPL stays unblocked.

An exclusive clear operation can also be performed for non-revertive mode and revertive mode using the ERPv2 CLI to clear any pending state.

Untagged Service VLAN

R-APS channel can be untagged by removing VLAN type configuration check on the Service VLAN (SVLAN).

When specifying a SVLAN, the configuration must check that the ring port(s) are members of this VLAN, tagged or untagged.

The VLAN and VPAs must be created first.

Note. All the nodes and ring ports must be configured with the same default or untagged VLAN.

Example: To configure an untagged R-APS channel for ring 1

On all nodes, a default or untagged VLAN must be configured on the ring ports:

```
-> vlan 4000
-> vlan 4000 members port 1/1-2 untagged
-> erp-ring 1 port1 1/1 port2 1/2 service-vlan 4000 level 2
```

Interaction With Other Features

This section contains important information about interaction of ERP with other OmniSwitch features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

Multicast

- IP multicast switching (IPMS) treats the ERP Service VLAN the same as any other configured VLAN on the switch. The ERP Service VLAN may carry data traffic, and if enabled and configured to do so, IPMS will perform regular multicast snooping on that VLAN.
- Disabling IPMS on the ERP Service VLAN is recommended if IP multicast routing or multicast snooping is enabled for the switch.

Spanning Tree

- Spanning Tree is automatically disabled when ERP is enabled on any port.
- On a switch running AOS Release 6 (for example, an OmniSwitch 6450), the default VLAN for ERP ports is protected and controlled by Spanning Tree.
- On a switch running AOS Release 7 or 8 (for example, an OmniSwitch 6900), the default VLAN for ERP ports is protected and controlled by ERP.

VLAN Stacking

ERP has the following interactions with VLAN Stacking:

- ERP is supported on Network Network Interface (NNI) ports; it is not supported on UNI ports.
- Tunneling of STP BPDUs across UNI ports is supported in a VLAN stacking configuration.

See "Configuring ERP with VLAN Stacking NNIs" on page 12-15 for more information.

Source Learning

The ERP protocol determines and performs the MAC address flushing per port.

QoS Interface

The interaction between ERP and QoS is for the purpose so that R-APS PDUs can be handled appropriately by the switch.

MVRP

ERP NI must provide blocking or forwarding state of ERP ports to MVRP.

Quick Steps for Configuring ERP with Standard VLANs

The following steps provide a quick tutorial for configuring ERP.

1 Create a VLAN using the **vlan** command and add the ring ports.

```
-> vlan 1001
-> vlan 1001 members port 1/1-2 tagged
```

2 Create ERP ring ID 1, ERP Service VLAN and MEG Level and associate two ports to the ring using the **erp-ring** command.

```
-> erp-ring 1 port1 1/1 port2 1/2 service-vlan 1001 level 1
```

3 Configure the RPL on one node using the **erp-ring rpl-node** command.

```
-> erp-ring 1 rpl-node port 1/1
```

4 Create additional VLANs and add to the ring ports using the **vlan** command.

```
-> vlan 11-20
-> vlan 11-20 members port 1/1-2 tagged
```

5 Enable the ERP ring configuration using the **erp-ring enable** command.

```
-> erp-ring 1 enable
```

6 Display the ERP configuration using the **show erp** command.

```
-> show erp
```

Quick Steps for Configuring ERP with VLAN Stacking

The following steps provide a quick tutorial for configuring ERP with VLAN Stacking:

1 Create a VLAN Stacking SVLAN 1001 using the command.

```
-> ethernet-service svlan 1001
```

2 Create a VLAN Stacking service and associate the service with SVLAN 1001 using the **ethernet-service service-name** command.

```
-> ethernet-service service-name CustomerA svlan 1001
```

3 Configure ports 1/1 and 1/2 as VLAN Stacking Network Network Interface (NNI) ports, associate the ports with SVLAN 1001, and configure them for use with ERP using the **ethernet-service svlan nni** command.

```
-> ethernet-service nni port 1/1
-> ethernet-service nni port 1/2
-> ethernet-service svlan 1001 nni port 1/1
-> ethernet-service svlan 1001 nni port 1/2
```

4 Create ERP ring ID 1 and associate the two NNI ports to the ring using the **erp-ring** command.

```
-> erp-ring 1 port1 1/1 port2 1/2 service-vlan 1001 level 5
```

5 Configure the RPL on one node using the **erp-ring rpl-node** command.

```
-> erp-ring 1 rpl-node port 1/1
```

6 Create additional SVLANs and add to the ring ports using the command.

```
-> ethernet-service svlan 1002
-> ethernet-service svlan 1003
-> ethernet-service svlan 1002 nni port 1/1-2
-> ethernet-service svlan 1002 nni port 1/2-2
```

7 Enable the ERP ring configuration using the **erp-ring enable** command.

```
-> erp-ring 1 enable
```

8 Display the ERP configuration using the **show erp** command.

```
-> show erp
```

ERP Configuration Overview and Guidelines

Configuring ERP requires several steps. These steps are outlined here and further described throughout this section. For a brief tutorial on configuring ERP, see ““Quick Steps for Configuring ERP with Standard VLANs” on page 12-10.

By default, ERP is disabled on a switch. Configuring ERP consists of these main tasks:

1 Configure the basic components of an ERP ring (ring ports, service VLAN, and MEG level). See “Configuring an ERP Ring” on page 12-13.

2 Tag VLANs for ring protection. See “Adding VLANs to Ring Ports” on page 12-13.

3 Configure an RPL port. When a ring port is configured as an RPL port, the node to which the port belongs becomes the RPL owner. The RPL owner is responsible for blocking and unblocking the RPL. See “Configuring an RPL Port” on page 12-14.

4 Change the Wait-To-Restore timer value. This timer value determines how long the RPL owner waits before restoring the RPL to a forwarding state. See “Setting the Wait-to-Restore Timer” on page 12-14.

5 Change the Guard timer value. This timer value determines an amount of time during which ring nodes ignore R-APS messages. See “Setting the Guard Timer” on page 12-14.

6 Configure the ring port to receive the loss of connectivity event for a Remote Ethernet OAM endpoint. See “Configuring ERP with VLAN Stacking NNIs” on page 12-15.

7 Configure a VLAN Stacking NNI-to-SVLAN association for ERP control. This is done to include an SVLAN in a ring configuration. See “Configuring ERP with VLAN Stacking NNIs” on page 12-15.

8 Clear ERP statistics. Commands to clear ERP statistics for a single ring or multiple rings are described in “Clearing ERP Statistics” on page 12-16.

Configuration Guidelines

Use the following guidelines when configuring ERP for the switch:

- Physical switch ports and logical link aggregate ports can be configured as ERP ring ports. This also includes VLAN Stacking Network Network Interface (NNI) ports.
- ERP is *not* supported on mobile ports, mirroring ports, link aggregate member ports, multicast VLAN receiver ports (ERP is supported on Multicast VLAN sender ports only), or VLAN Stacking User Network Interface (UNI) ports.
- An ERP ring port can belong to only one ERP ring at a time.
- STP is automatically disabled when ERP is enabled on any port.
- If the ERP switch participates in an Ethernet OAM Maintenance Domain (MD), configure the Management Entity Group (MEG) level of the ERP service VLAN with the number that is used for the Ethernet OAM MD.
- The Service VLAN can belong to only one ERP ring at a time and must be a static VLAN. Note that the service VLAN is also a protected VLAN.

Configuring an ERP Ring

The following configuration steps are required to create an ERP ring:

1 Determine which two ports on the switch are the ring ports. For example, ports 1/1 and 1/2.

2 Determine which VLAN on the switch is the ERP service VLAN for the ring. If the VLAN does not exist, create the VLAN. And add the ring ports to the VLAN as tagged or untagged. For example:

```
-> vlan 500
-> vlan 500 members port 1/1-2 tagged
```

3 Create the ERP ring configuration on each switch using the **erp-ring** command. For example the following command configures an ERP ring with ring ID 1 on ports 1/2 and 1/2 along with service VLAN 500 and MEG level 1.

```
-> erp-ring 1 port1 1/1 port2 1/2 service-vlan 500 level 1
-> erp-ring 1 enable
```

To configure link aggregate logical ports as ring ports, use the **erp-ring** command with the **linkagg** parameter. For example:

```
-> erp-ring 1 port1 linkagg 1 port2 linkagg 2 service-vlan 500 level 1
-> erp-ring 1 enable
```

4 Repeat Steps 1 through 6 for each switch that participates in the ERP ring. Make sure to use the same VLAN ID and MEG level for the service VLAN on each switch.

Use the **show erp** command to verify the ERP ring configuration. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Removing an ERP Ring

To delete an ERP ring from the switch configuration, use the **no** form of the **erp-ring** command. For example:

```
-> no erp-ring 1
```

Note. Administratively disable ring ports before deleting the ring to avoid creating any network loops. Once a ring is deleted, then administratively enable the ports under Spanning Tree protocol.

Adding VLANs to Ring Ports

ERP allows a single VLAN or a number of VLANs to participate in a single ERP ring. The **vlan members tagged** command is used to tag the ring ports of the ERP ring with a VLAN ID.

To add a VLAN or range of VLANs to ring ports use the **vlan members tagged** command**.**

```
-> vlan 12-20
-> vlan 12-20 members port 1/1 tagged
-> vlan 12-20 members port 1/2 tagged
```

Configuring an RPL Port

A ring protection link (RPL) port can be a physical or logical port. The port must be a ring port before it is configured as an RPL port, and out of the two ring ports on the node, only one can be configured as a RPL port. The RPL remains blocked to prevent loops within the ERP ring.

To configure an RPL port, first disable the ring and then use the **erp-ring rpl-node** command to specify which ring port serves as the RPL. For example:

```
-> erp-ring 1 disable
-> erp-ring 1 rpl-node port 1/1
-> erp-ring 1 enable
```

Note. RPL node can be configured only when the ring is disabled; RPL configuration applied to the ring while it is enabled is rejected.

To remove the RPL node configuration for the specified ring, use the **no** form of the **erp-ring rpl-node** command. For example:

```
-> no erp-ring 1 rpl-node
```

To verify the RPL node configuration for the switch, use the **show erp** command. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Setting the Wait-to-Restore Timer

The wait-to-restore (WTR) timer determines the number of minutes the RPL owner waits before blocking the RPL port after the ERP ring has recovered from a link failure.

By default, the WTR time is set to five minutes. To change the value of the WTR timer, use the **erp-ring wait-to-restore** command. For example:

```
-> erp-ring 1 wait-to-restore 6
```

The above command is only used on a switch that serves as the RPL node for the ERP ring. The specified ERP ring ID must already exist in the switch configuration.

To restore the timer back to the default setting, use the **no** form of the **erp-ring wait-to-restore** command. For example:

```
-> no erp-ring 1 wait-to-restore
```

To verify the WTR configuration, use the **show erp** command. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Setting the Guard Timer

The guard timer is used to prevent the ring nodes from receiving outdated R-APS messages, which are no longer relevant. Receiving outdated R-APS messages could result in incorrect switching decisions. During the amount of time determined by this timer, all received R-APS messages are ignored by the ring protection control process.

By default, the guard timer value is set to 50 centi-seconds. To change the value of this timer, use the **erp-ring guard-timer** command. For example:

```
-> erp-ring 1 guard-timer 100
```

To restore the Guard Timer back to the default value, use the no form of the erp-ring guard-timer command. For example:

```
-> no erp-ring 1 guard-timer
```

To verify the configured Guard Timer, use the **show erp** command. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring ERP with VLAN Stacking NNIs

A VLAN Stacking Network Network Interface (NNI) can participate in an ERP ring. However, an NNI is created through an association of a port with an SVLAN. Both STP and ERP cannot control the same VLAN-port association (VPA). By default, the NNI to SVLAN association is controlled by STP.

To include an NNI in an ERP ring, specify ERP control at the time the NNI association is configured. This is done using the **erp** parameter of the **ethernet-service svlan nni** command. For example:

```
-> ethernet-service svlan 1001 nni port 1/1
-> ethernet-service svlan 1001 nni port 1/2
```

The above commands configure ports 1/1 and 1/2 as NNI ports for SVLAN 1001. Note that the SVLAN specified must already exist in the switch configuration.

To configure an ERP ring with NNI-SVLAN associations, use the **erp-ring** command but specify an SVLAN ID for the service VLAN and the associated NNI ports as the ring ports. For example:

```
-> erp-ring 1 port1 1/1 port2 1/2 service-vlan 1001 level 2
-> erp-ring 1 enable
```

Note the following when configuring an ERP ring with VLAN Stacking NNI-SVLAN associations:

- Only two ERP type NNI associations are allowed per SVLAN.
- Configuring an ERP ring on 802.1q tagged port associations with SVLANs is not allowed.
- Configuring an ERP Ring on an STP type NNI association with an SVLAN is not allowed.
- Configuring an IMPVLAN as an ERP service VLAN is not allowed.
- If an SVLAN that is not associated with any NNI ports is configured as the service VLAN for an ERP ring, the NNI ring ports are automatically associated with that SVLAN at the time the ring is created.
- SVLAN User Network Interface (UNI) associations are not eligible for ERP ring protection.
- If the ERP type NNI ports are connected to the STP path through UNI ports, then STP BPDUs can be tunneled with the help of VLAN-stacking mechanism.
- Deleting an ERP service VLAN and it is associated NNI ports is only allowed when the ERP ring itself is deleted using the **no** for of the **erp-ring** command. None of the VLAN Stacking CLI commands can remove a service VLAN consisting of an NNI-SVLAN association.

Configuring ERP Protected SVLANs

An SVLAN becomes an ERP protected SVLAN when the SVLAN is associated with two NNI ports that also serve as ring ports. In this case, the SVLAN is automatically protected as part of the association with NNI ring ports.

The following sequence of configuration commands provides an example of how SVLANs are automatically added as protected SVLANs to an ERP ring:

```
-> ethernet-service svlan 100
-> ethernet-service svlan 200
-> ethernet-service svlan 300
-> ethernet-service svlan 400
-> ethernet-service svlan 100 nni port 1/1-2
-> ethernet-service svlan 200 nni port 1/1-2
-> ethernet-service svlan 300 nni port 1/1-2
-> erp-ring 10 port1 1/1 port 2 1/2 service-vlan 400 level 1
```

In the above example:

- SVLANs 100, 200, and 300 are automatically added as protected VLANs when the ring is created. This is due to the NNI ports being part of ERP ring 10.
- SVLAN 400 is also automatically added as a protected VLAN when it is configured as the service VLAN for the ring.

Use the **show erp** command to verify the configured VLAN Stacking ERP ring configuration. For more information about these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Clearing ERP Statistics

To clear ERP statistics for all rings in the switch, use the **clear erp statistics** command. For example:

```
-> clear erp statistics
```

To clear ERP statistics for a specific ring in the switch, use the **clear erp statistics** command with the **ring** parameter to specify a ring ID. For example:

```
-> clear erp statistics ring 5
```

To clear ERP statistics for a specific ring port, use the **clear erp statistics** command with the **ring** and **port** parameters. For example:

```
-> clear erp statistics ring 5 port 1/2
```

To clear ERP statistics for a specific link aggregate ring port, use **clear erp statistics** command with the ring and **linkagg** parameters. For example:

```
-> clear erp statistics ring 5 linkagg 2
```

Use the **show erp statistics** command to verify ERP statistics. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

ERPv2 Configuration Overview and Guidelines

The following section details the guidelines and prerequisites for configuring ERPv2 and details on how to configure the ERPv2 related parameters using the OmniSwitch CLI. Configuring the sample ERPv2 ring network involves the following tasks:

1 *Optional:* Configure tagged ports or link aggregate ports before configuring ERP.

2 Configure an ERP ring with same ERP ring ID on all switches in the network.

3 Define same ERP Service VLAN on all switches.

4 Set the same Management Entity Group (MEG) (for example, level 2) for all switches.

5 Assign one switch to be the RPL owner. Configure the port connected to the Ring Protection Link as an RPL port.

6 Enable the configured ERPv2 ring.

7 Assign separate VLANs as protected VLANs to the ERP ring.

8 Use the default settings for the guard timer and WTR timer values. These values can be adjusted as necessary.

The following sub-sections provide the details on prerequisites and different configurations for switches to set up an ERPv2 ring network using OmniSwitch CLI commands.

Major and Sub Ring Management

A shared link must be configured only on the major ring.

The following conditions must be considered for configuring an ERPv2 port for a shared link:

- Sub-rings can not be closed using a shared link.
- An SVLAN must exist before an ERP ring is created and must be unique per ring.
- A given port can only be configured on one ring.
- Each ring must have its own RPL.
- The RPL can be placed anywhere on the major ring including the shared links.
- The RPL can be placed anywhere on the sub-rings, including the sub-ring-port. Since the sub-ring is not closed using the shared link, the RPL cannot be placed on the shared link.

Configuration Parameters

The following conditions must be considered before configuring an ERPv2 port:

- A given port can only be configured on one ring.
- The shared links are only configurable on the Master Ring.
- The Sub Rings cannot be closed using the shared links.
- Each ring must have its own RPL.

- The RPL can be placed anywhere on the Master Ring, including the shared links.
- The RPL can be placed anywhere on the Sub Rings, including the only ring port of the interconnection nodes. Since the sub-ring is not closed using the shared link, the RPL cannot be placed on the shared link.

ERPv2 Ring Sample Configuration

A master ring can be configured using the following command:

```
Switch 1-> erp-ring 1 port1 1/1 port2 1/2 service-vlan 10 level 2
```

A sub-ring on the non-interconnection node can be configured using the following command:

```
Switch 2-> erp-ring 2 port1 1/1 port2 1/3 service-vlan 10 level 2
```

A sub ring on the interconnection node can be configured using the following command:

```
Switch 3-> erp-ring 3 sub-ring-port 1/3 service-vlan 10 level 2
```

Sample Switch Configuration

The following configurations must be performed on each switch in the ERPv2 Ring network:

Step 1 : Create the Service VLAN and add to ring ports.

```
-> vlan 10
-> vlan 200
-> vlan 10 members port 1/3 tagged
-> vlan 10 members port 1/5 tagged
-> vlan 200 members port 1/6 tagged
```

Step 2 : Create the rings.

```
-> erp-ring 1 port1 1/5 port2 1/3 service-vlan 10 level 1
-> erp-ring 2 sub-ring-port 1/6 service-vlan 200 level 1
```

Step 3 : Create traffic VLANs and add to ring ports as necessary using VM commands

```
-> vlan 100-400
-> vlan 100-300 members port 1/5 tagged
-> vlan 100-300 members port 1/3 tagged
-> vlan 201-400 members port 1/6 tagged
```

Step 4 : Enable the rings.

```
-> erp-ring 1 enable
-> erp-ring 2 enable
```

Note. The traffic VLANs could be added or deleted as needed at any time during the configuration.

Enabling and Disabling R-APS Virtual Channel

User can enable and disable virtual channel. By default, R-APS virtual channel is enabled.

Enabling R-APS Virtual Channel

Enable R-APS virtual channel using the following command:

```
-> erp-ring 2 virtual-channel enable
```

R-APS messages from the sub-ring on the interconnection node are forwarded as normal data to the major ring ports. A node is identified as interconnection node when at least one ring is configured with a sub-ring-port.

R-APS messages from the sub-ring are tagged with the sub-ring SVLAN, are forwarded to the major ring member ports of this SVLAN.

Note. All the ring ports in major ring must be member of the sub-ring SVLAN to support R-APS virtual channel.

Interconnection Node of the Sub-Ring

When R-APS virtual channel is enabled, on the interconnection node of a sub-ring, all the R-APS messages received from sub-ring port are processed and flooded to major ring ports that are the member of the VLAN used by R-APS message. For example,

```
-> erp-ring 3 virtual-channel enable
```

Other nodes of the Sub-Ring

When enabled, R-APS messages received on blocked port are processed but not forwarded to the other ring port.

Disabling R-APS Virtual Channel

Disable R-APS virtual channel using the following command:

```
-> erp-ring 2 virtual-channel disable
```

Now, R-APS messages from the sub-ring on the interconnection node are not forwarded to any other ports. R-APS messages are forwarded even on the blocked ports in the sub-ring. A configuration object is required for the sub-ring to disable the R-APS virtual channel.

Interconnection Node of the Sub-Ring

When virtual channel is disabled, R-APS message received from sub-ring ports are processed but not flooded to major ring. For example,

```
-> erp-ring 3 virtual-channel disable
```

Other nodes of the Sub-Ring

When virtual channel is disabled, R-APS messages received on blocked port are processed and forwarded to other ring port.

Note. Virtual channel configuration must be consistent among all nodes of the sub-ring.

Enabling or Disabling Revertive Mode

Revertive mode is enabled by default. You can disable revertive mode by setting the following command:

```
-> erp-ring 2 revertive enable
```

You can enable revertive mode by setting following command:

```
-> erp-ring 2 revertive disable
```

Non-revertive Mode

Under non-revertive mode, when the failure recovers, the blocked port stays blocked and the unblocked RPL stays unblocked. Revertive mode is enabled by default. Operator can enable non-revertive mode by setting following command.

When the ERPv2 node is operating with ERPv1 node in the same ring, it operates in different way for compatibility. In this mode, revertive mode is always assumed, it operates in revertive mode regardless of user configuration.

```
-> erp-ring 2 revertive disable
```

Clear Non-revertive and Revertive Mode

When the ring is in the No Request (NR) state and the blocked port is not the RPL port, the operator must be allowed to trigger the reversion to the initial state of the ring (make the RPL port blocked).

This situation happens in 2 cases:

- The ring is set in a non-revertive mode.
- The ring is set in a revertive mode but the WTR timer has not expired.

The CLI command is as follows:

```
-> erp-ring 2 clear
```

The command can only be issued on the RPL owner node and when the ring is in the NR state and WTR timer not expired or no WTR (non-revertive mode)

When the command is accepted, the RPL owner node blocks its RPL port, and transmits an R-APS (NR, RB) message in both directions. Upon receiving the R-APS (NR, RB), each node unblocks its blocking ports and performs a flush operation when applicable.

Sample Ethernet Ring Protection Configuration

This section provides an example network configuration in which ERP is configured on network switches to maintain a loop-free topology. In addition, a tutorial is also included that provides steps on how to configure the example network topology using the Command Line Interface (CLI).

Example ERP Overview

The following diagram shows a five-switch ERP ring configuration:

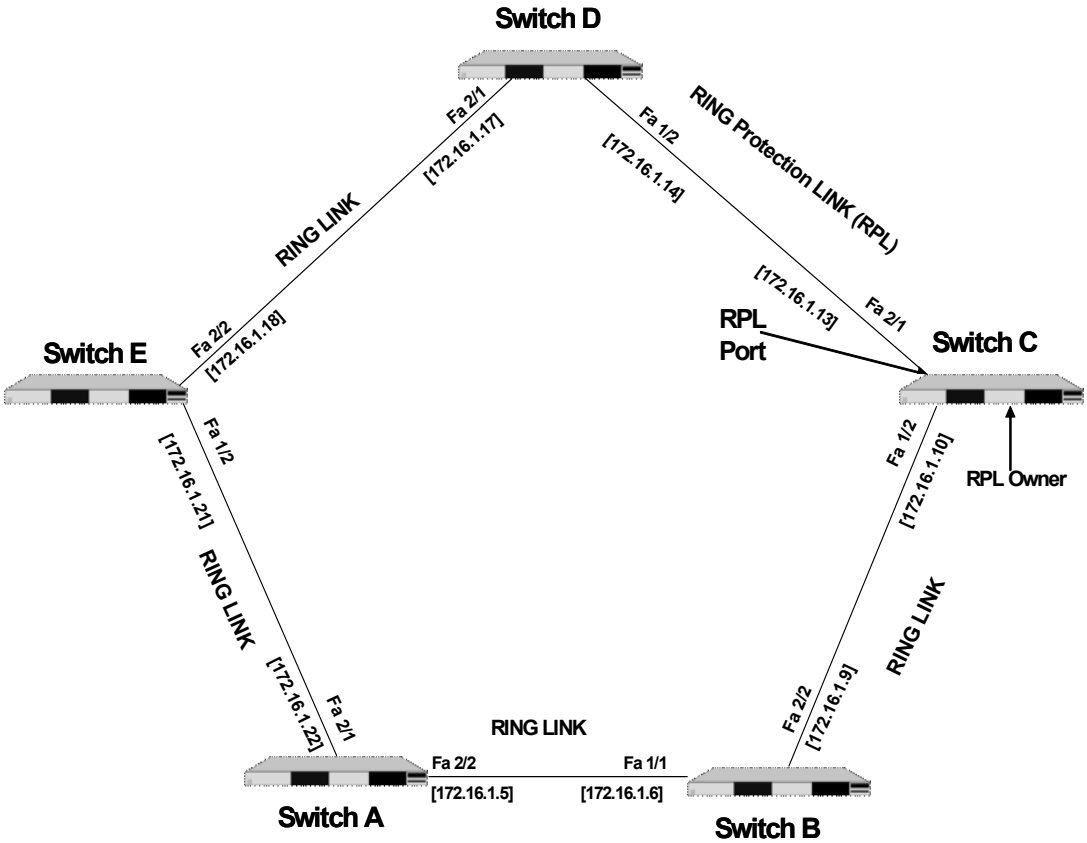


Figure 12-4 :Example ERP Overview

Configuring the sample ERP ring network shown in the above diagram involves the following tasks:

1 Configure an ERP ring with ERP ring ID 1 on all switches in the network.

2 Define an ERP Service VLAN as VLAN 10 on all switches.

3 Set the Management Entity Group (MEG) level to 2 for all switches.

4 Switch C is the RPL owner; configure the port connected to the Ring Protection Link as a RPL port.

5 Enable the configured ERP ring.

6 Assign VLANs 11-20 as a protected VLANs to ERP ring 1.

7 Use the default settings for the guard timer and WTR timer values. These values can be adjusted as necessary.

Example ERP Configuration Steps

The following steps provide a quick tutorial for configuring the ERP ring network shown in the diagram on page 12-21:

1 Configure ERP ring 1 and add protected VLANs 11 through 20 on Switch A, B, C, D, and E using the following commands:

```
-> vlan 10
-> vlan 10 members port 2/1-2 tagged
-> erp-ring 1 port1 2/1 port2 2/2 service-vlan 10 level 2
-> erp-ring 1 enable
-> vlan 11-20 members port 2/1-2 tagged
```

2 Configure Switch C as the RPL owner for the ring using the following commands to designate port 2/1 as the RPL port:

```
-> erp-ring 1 disable
-> erp-ring 1 rpl-node port 2/1
-> erp-ring 1 enable
```

3 Verify the ERP ring configuration on any switch using the following command:

```
-> show erp ring 1
Legend: *   to Inactive Configuration
Ring Id                 : 1,
Ring Type               : Normal Ring,
Ring Port1              : 1/1/3,
Ring Port2              : 1/1/4,
Ring Status             : enabled,
Service VLAN            : 10,
Revertive Mode          : enable,
WTR Timer (min)         : 5,
Guard Timer (centi-sec) : 50,
Virtual Channel         : enable,
MEG Level               : 1,
Ring State              : protection,
Active ERP version      : Ver 2,
Ring Node Type          : non-rpl,
Ring Profile            : SAP-SAP,
Last State Change       : Mon Jun 29 2020 22:21:49  (sysUpTime 2d:06h:41m)
```

The above output example shows that ERP ring 1 is created on ring ports 2/1 and 1/2 with service VLAN 10, WTR timer of 5 mins, Guard timer of 50 centi-seconds, MEG level 2, and port 2/1 is the RPL port.

4 Verify the status of an ERP ring port on any switch using the following command:

```
-> show erp port 1/1/2
Ring-Id : 1
  Ring Port Status     : blocking,
  Ring Port Type       : non-rpl,
  Ethoam Event         : disabled,
  Remote-endpoint Id   : none
  SPB-Remote-system Id : 11:22:33:44:55:66
  SPB SAP Neighbor     : no
```

The above command shows the forwarding status of the port, the type of ring port (RPL or non-RPL), and ETHOAM event status.

Sample ERPv2 Ring Configuration

This section provides an example network configuration in which ERPv2 is configured on network switches to maintain a loop-free topology. In addition, a tutorial is also included that provides steps on how to configure the example network topology using the Command Line Interface (CLI).

Example ERPv2 Overview

The following diagram shows a seven-switch ERPv2 ring configuration when R-APS virtual channel is enabled.

The topology of the network is as follows:

- Switches A, B, C, D, and E for the Major Ring.
- Switch A and B form a shared link.
- Switch C is configured to be the main RPL node.
- Switches A, B, F, and G form the Sub Ring.

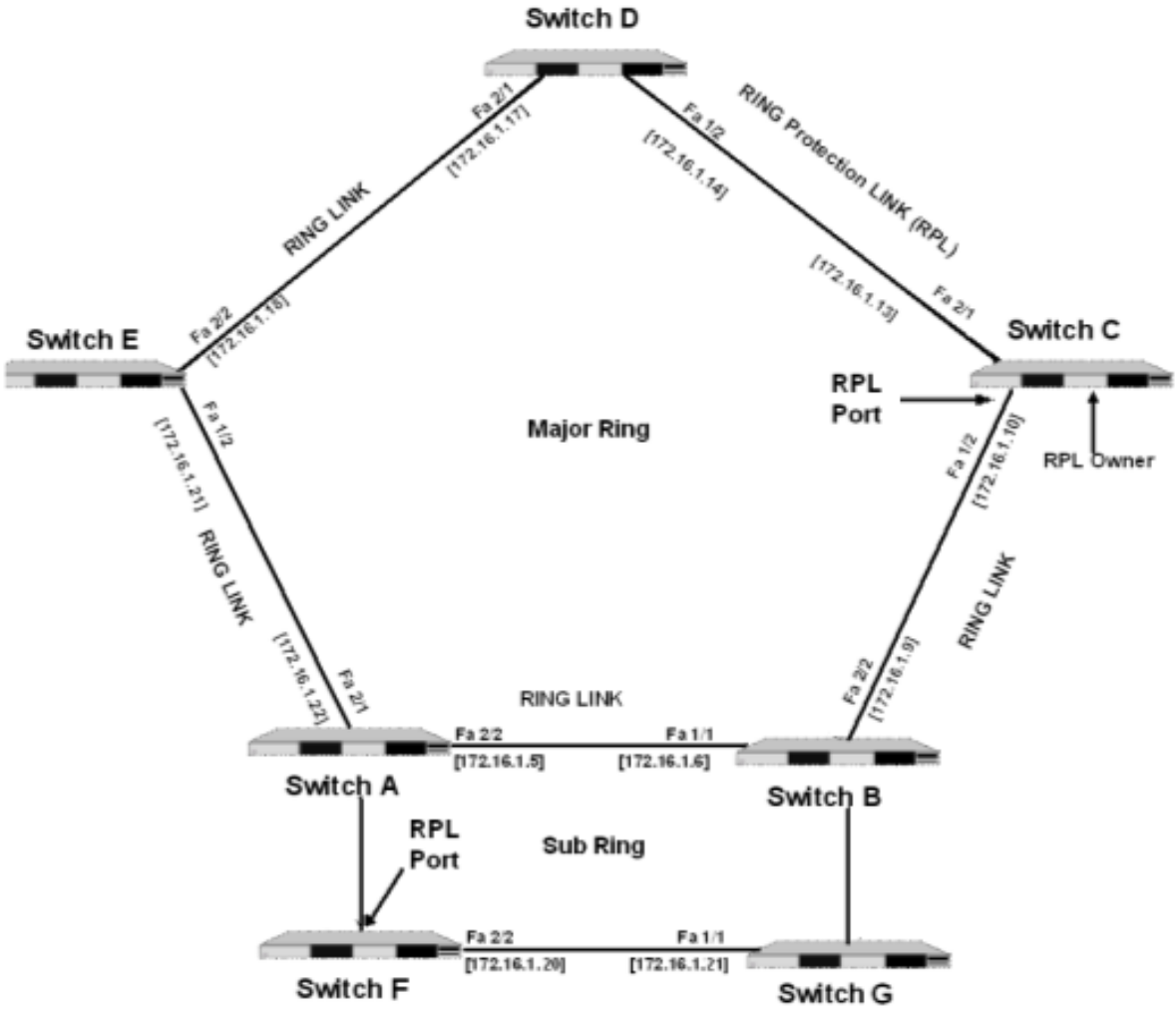


Figure 12-5 :Example ERPv2 Overview

The following sub-sections provide the details on prerequisites and different configurations for switches to set up an ERPv2 ring network using OmniSwitch CLI commands.

Configuring Shared Link

The following configurations must be performed on Switch A and Switch B.

Step 1 : Create the Service VLAN and add to ring ports on Switch A and B that are part of a shared link:

```
Switch A -> vlan 10
Switch A -> vlan 200
Switch A -> vlan 10 members port 2/1 tagged
Switch A -> vlan 10 members port 2/2 tagged
Switch A -> vlan 200 members port 1/6 tagged

Switch B -> vlan 10
Switch B -> vlan 200
Switch B -> vlan 10 members port 1/1 tagged
Switch B -> vlan 10 members port 2/2 tagged
Switch B -> vlan 200 members port 1/6 tagged
```

Step 2 : Create the ERP rings 1 and 2 on Switch A.

```
Switch A -> erp-ring 1 port1 2/1 port2 2/2 service-vlan 10 level 1
Switch A -> erp-ring 2 sub-ring-port 1/6 service-vlan 200 level 1
```

Step 3 : Create traffic VLANs and add to ring ports as necessary using VM commands on Switch A.

```
Switch A -> vlan 100-400
Switch A -> vlan 100-300 members port 2/1 tagged
Switch A -> vlan 100-300 members port 2/2 tagged
Switch A -> vlan 201-400 members port 1/6 tagged
```

Step 4 : Enable the rings on Switch A.

```
Switch A -> erp-ring 1 enable
Switch A -> erp-ring 2 enable
```

Configuring Main RPL Node

Main RPL is configured on the Switch B. The following configurations must be performed on Switch B.

Step 1 : Create the ERP rings 1 and 2 on Switch B.

```
Switch B -> erp-ring 1 port1 1/1 port2 2/2 service-vlan 10 level 1
Switch B -> erp-ring 2 sub-ring-port 1/6 service-vlan 2000 level 1
```

Step 2 : Configure Switch B as RPL Node using the **erp-ring rpl-node** command:

```
Switch B -> erp-ring 1 rpl-node 2/2
```

Step 3 : Enable the rings on Switch B.

```
Switch B -> erp-ring 1 enable
Switch B -> erp-ring 2 enable
```

Step 4 : Create traffic VLANs and add to ring ports as necessary Switch B.

```
Switch B -> vlan 100-400
Switch B -> vlan 100-300 members port 1/1 tagged
```

```
Switch B -> vlan 100-300 members port 2/2 tagged
Switch B -> vlan 201-400 members port 1/6 tagged
```

Configuring Switches in Main Ring

The following configurations must be performed on Switch C, D, and E

```
-> vlan 10
-> vlan 10 members port 1/2 tagged
-> vlan 10 members port 2/1 tagged
-> erp-ring 1 port1 1/2 port2 2/1 service-vlan 10 level 1
-> vlan 100-300
-> erp-ring 1 enable
-> vlan 100-300 members port 1/2 tagged
-> vlan 100-300 members port 2/1 tagged
```

Configuring Secondary RPL Node

The following configurations must be performed on Switch F in the ERPv2 Ring network:

```
-> vlan 200-400
-> vlan 200-400 members port 1/6 tagged
-> vlan 200-400 members port 2/2 tagged
-> erp-ring 2 port1 1/6 port2 2/2 service-vlan 200 level 1
-> erp-ring 2 rpl-node 1/6
-> erp-ring 2 enable
```

Configuring Switch in Sub Ring

The following configurations must be performed on Switch G in the ERPv2 Ring network:

```
-> vlan 200-400
-> vlan 200-400 members port 1/1 tagged
-> vlan 200-400 members port 1/6 tagged
-> erp-ring 2 port1 1/1 port2 1/6 service-vlan 200 level 1
-> erp-ring 2 enable
```

Configuring ERP over SPB

This feature is to allow seamless connectivity between an access ERP ring and an SPBM aggregation network by enabling ERP in the BEB of the SPB domain. The feature allows ERP protected VLANs to be mapped dynamically and manually to a service on SPBM network on the same SAP.

This feature supports the following on a gateway switch which supports both ERP and SPBM.

- Access port and Access port configuration - An access ERP ring connecting to a single SPBM backbone (when both sides of the ERP ring end at the same BEB of the SPB domain).
- Access port and SDP/tunnel configuration - An ERP ring using an intermediate SPBM network as transport, access ERP ring connected to two different BEBs on the SPBM backbone (ERP ring might end on two different BEBs connected through the SPMB cloud.)

Some points to note:

- Only two ERP type NNI associations are allowed per SVLAN.
- Configuring an ERP ring on 802.1q tagged port associations with SVLANs is not allowed.
- Configuring an ERP ring on an STP type NNI association with an SVLAN is not allowed.
- BEB cannot be a RPL node. Hence, RPL port shall not be configured on SPB network. The RPL port cannot be configured as a SAP neighbor.
- For the SPB Service associated with the ERP Service VLAN, it has to be configured in the Control bridge VLAN (BVLAN) of the SPB network. This will ensure reachability to all nodes of the SPB network.
- In case where the underlay network needs to support more than one ERP ring, then
  - Ensure there is no overlap in the VLAN range supported within the ERP rings. That is, each ERP ring must have an exclusive range of VLANs including the service VLAN relative to the other ERP rings.
- In the underlay network, the services association with the ERP VLANS is exclusive to each ring. That is, the service IDs (for example, ISID, VNID) cannot extend across/into the other ERP rings.

Access Port and Access Port Configuration

Create the ERP ring configuration on each switch using the **erp-ring** command on the Layer 2 domain. For example the following command configures an ERP ring with ring ID 1 on linkagg 31 and 32 along with service VLAN 500 and MEG level 1.

```
-> erp-ring 50 port1 linkagg 31 port2 linkagg 32 service-vlan 500 level 1
-> erp-ring 50 enable
```

If the ERP service VLAN (control VLAN) is mapped to tagged SAP, configure the ERP ring as **access-tagged** using the **erp-ring** command as following on the SPB domain, with the other required SAP configurations.

```
-> service access linkagg 31
-> service 1001 spb isid 1001 bvlan 4001
-> service 1001 sap linkagg 31:500

-> service access linkagg 41
-> service 1001 spb isid 1001 bvlan 4001
```

```
-> service 1001 sap linkagg 41:500

-> erp-ring 1 port1 access-tagged linkagg 31 port2 access-tagged linkagg 41
service-vlan 500 level 1
```

Similarly, if the service VLAN is mapped to untagged SAP, configure the ERP ring as **access-untagged** using the **erp-ring** command as following on the SPB domain.

```
-> service 1001 sap linkagg 31:0
-> service 1001 sap linkagg 41:0

-> erp-ring 1 port1 access-untagged linkagg 31 port2 access-untagged linkagg 41
service-vlan 10 level 1
```

Access Port and SDP/Tunnel Configuration

Create the ERP ring configuration on each switch using the **erp-ring** command on the Layer 2 domain. For example the following command configures an ERP ring with ring ID 1 on linkagg 31 and 32 along with service VLAN 500 and MEG level 1.

```
-> erp-ring 50 port1 linkagg 31 port2 linkagg 32 service-vlan 500 level 1
-> erp-ring 50 enable
```

Configure an ERP ring SAP neighbor by using **erp-ring sap-neighbor** command on the Layer 2 domain.

```
-> erp-ring 10 sap-neighbor linkagg 31
```

If the ERP service VLAN (control VLAN) is mapped to tagged SAP, configure the ERP ring as **access-tagged** using the **erp-ring** command as following on the SPB domain, with the other required SAP configurations.

```
-> service access linkagg 31
-> service 1001 spb isid 1001 bvlan 4001
-> service 1001 sap linkagg 31:500

-> service access linkagg 41
-> service 1001 spb isid 1001 bvlan 4001
-> service 1001 sap linkagg 41:500
```

For SAP-SDP configuration, in this example, linkagg 31 will be the access port that is connected to the ERP ring on the Layer 2 domain, and the second port will represent the MAC address of the remote system of the tunnel where the other side of the ERP ring is connected.

```
-> erp-ring 1 port1 access-tagged linkagg 31 spb remote-system 00:11:22:33:44:55
service-vlan 100 level 1
```

Access Port and Access Port Example

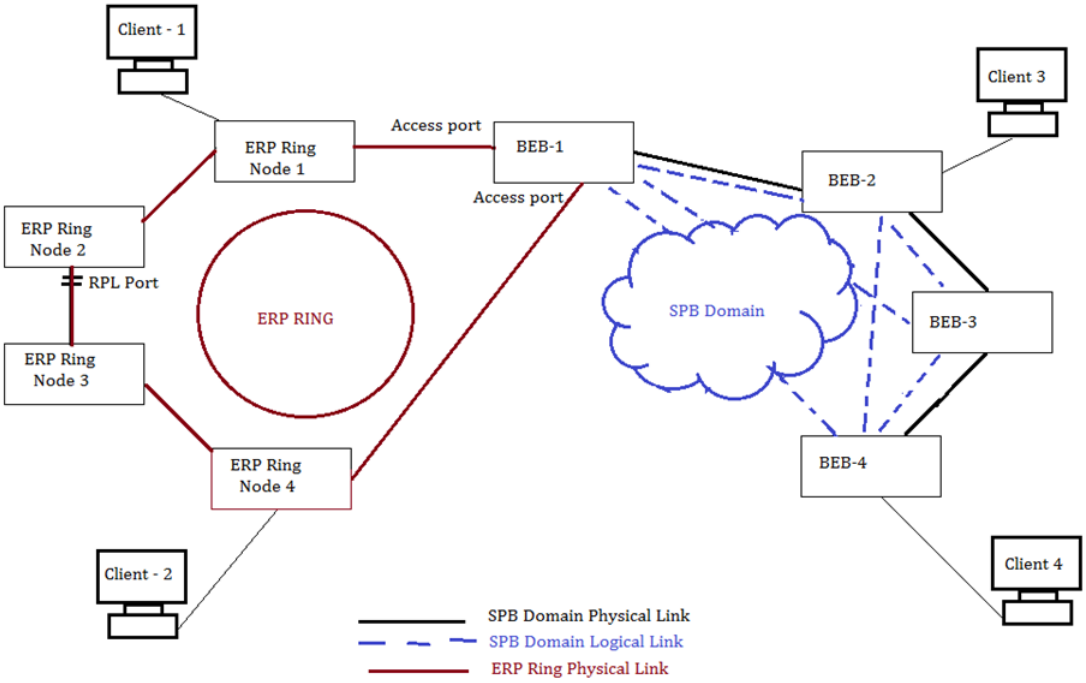


Figure 12-6 :Access Port and Access Port Example

In the given topology, ERP is be enabled on both the physical port or linkagg on the access ports. Upon a port failure, instead of blocking the port when the physical port goes down in VLAN domain, all the SAP/VP associated to the access port shall stay blocked until WTR expiry. Also flushing of the MAC address shall happen on the SAP/VP. Also when there is failure in the ERP access ring, MAC address on the SAP shall get flushed as per ERP protocol.

Access Port and SDP/Tunnel Example

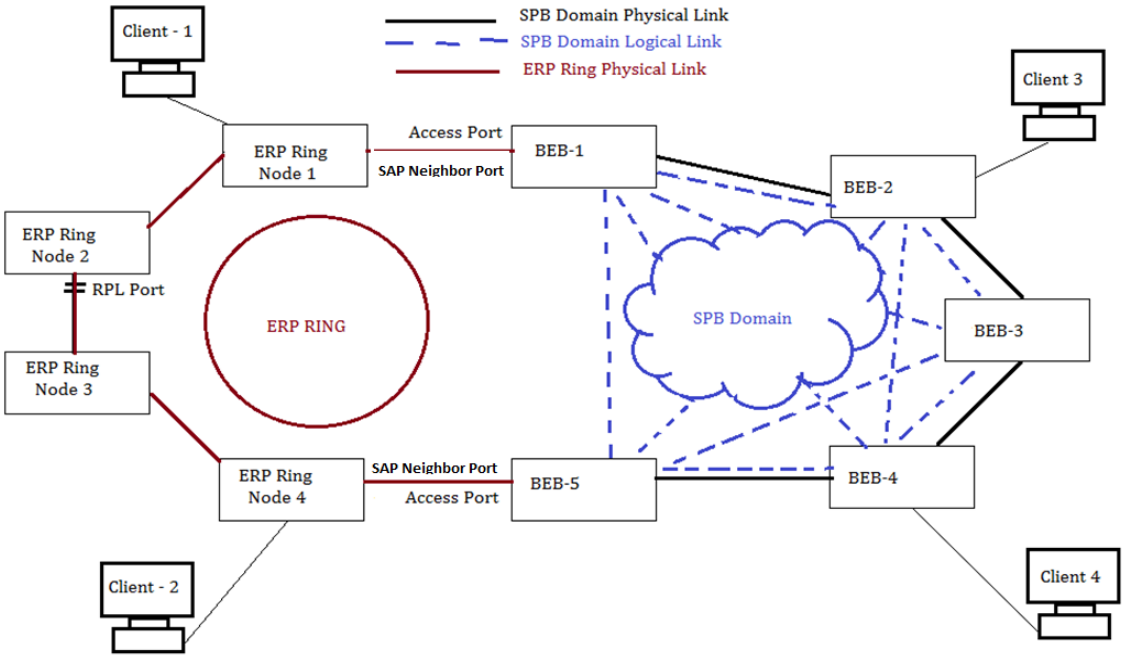


Figure 12-7 :Access Port and SDP/Tunnel Example

In this topology, ERP shall be configured on both BEB's (BEB-1 and BEB-5) that terminates the ERP ring. On both BEB's, one of the port or port1 shall be the access port, that is, connected to the ERP Ring and second port shall represent the remote system of the tunnel where the other side of the ERP ring is connected.

On ERP ring Node 1, SAP neighbor port is configured for the port that is adjacent to the SAP port of BEB-1. Similarly, on ERP ring Node 4, SAP neighbor port is configured for the port that is adjacent to the SAP port of BEB-5. The SAP neighbor cannot be configured as RPL port.

When there is a failure in the SBPM network/tunnel, ERP RPL node in the ERP ring shall be informed to unblock the RPL port by both the BEB's, but access port on both the end of the BEB's shall be kept forwarding so that client on the ERP ring shall have connectivity to the SPMB network.

Now, when the SPB network/Tunnel is up again, instead of blocking the port (in this case SDP/Tunnel) that comes up in regular ERP, until the WTR timer expiry, access port in one of the BEB (either on BEB-1 and BEB-5) based on the local system ID and user configured remote system ID shall be blocked and SBPM network/tunnel shall be provisioned. BEB with higher system ID shall block the access port. Access port also generates a flush since blocking the access port would trigger a topology change.

After the WTR timer expiry on the RPL node, BEB shall receive ERP message to unblock the port after which access port shall be unblocked.

Verifying the ERP Configuration

A summary of the **show** commands used for verifying the ERP configuration is given here:

show erp	Displays the ERP configuration information for all rings, a specific ring, or for a specific ring port.
show erp statistics	Displays the ERP statistics for all rings, a specific ring, or a specific ring port.
show ethernet-service	Displays configuration information for VLAN Stacking Ethernet services, which includes SVLANs and NNI port associations.
show ethernet-service nni	Displays the VLAN Stacking NNI configuration.
ethernet-service transparent-bridging	Displays a list of SVLANs configured for the switch.

For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

13 Configuring MRP

Media Redundancy Protocol (MRP) is an IEC standard that specifies a recovery protocol for use in high availability ring topology networks used in industrial automation networks. It is described in the IEC 62439-2. The protocol is designed to react to a single link or switch failure in the network and provide deterministic recovery time, which is important in ensuring real time data communication needs of industrial networks. During the failure of a connection, the media redundancy reconfigures the network so that the nodes can be accessed again through a substitute path.

In This Chapter

This chapter provides an overview about how MRP works and how to configure media redundancy parameters through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following information and configuration procedures are included in this chapter:

- "MRP Overview" on page 13-2
- "MRP Configuration Overview and Guidelines" on page 13-6
- "Verifying MRP Configuration" on page 13-8

MRP Overview

Media Redundancy Protocol (MRP) is designed to react deterministically on a single failure of an inter-switch link or switch in the ring or interconnection topology. A MRP compliant network shall have a ring topology with multiple nodes. During the failure of a connection, the media redundancy reconfigures the network so that the nodes can be accessed again through a substitute path.

The starting point for the ring redundancy method is a ring topology. Nodes of a ring topology can be switches or communication modules. MRP defines a Redundancy Manager (MRM) and Redundancy Clients (MRCs).

MRP Operation

In an MRP ring, one switch is the Media Redundancy Manager (MRM) and all others are the Media Redundancy Clients (MRC). The function of the MRM is to observe and to control the ring topology in order to react on network faults. The MRM does this by sending frames on one ring port over the ring and receiving them from the ring over its other ring port, and vice-versa in the other direction. One ring port of the MRM shall be connected to a ring port of an MRC. The other ring port of that MRC shall be connected to a ring port of another MRC or to the second ring port of the MRM, thereby forming a ring topology.

The other nodes in the ring have the role of Media Redundancy Clients (MRC). An MRC reacts on received reconfiguration frames from the MRM and can detect and signal link changes on its ring ports. Certain nodes or all nodes in the ring may also start as a Media Redundancy Auto-manager (MRA). MRAs select one MRM among each other by using a voting protocol. The remaining MRAs transition to the role of MRC. Each node in the ring is able to detect the failure or recovery of a neighboring node.

The MRM and the MRC shall have two ring ports. The MRM and MRC shall be able to detect the failure or recovery of a link on a ring port. All MRM and MRC ring ports support the following states:

- Disabled - ring ports drop all received frames.
- Blocked - ring ports drop all received frames except MRP control frames and some standard frames, for example, LLDP, IEEE 1588-2008 PTP peer to peer frames.
- Forwarding - All frames shall be passed through according to the forwarding behavior of IEEE 802.1D.

During normal operation, the network operates in the Ring-Closed state. To prevent a loop, one of the MRM ring ports is blocked, while the other port is forwarding. Most of the time, both ring ports of all MRCs are in the forwarding state. With this loop avoidance, the physical ring topology becomes a logical stub topology. In the event of any of the MRC link going down, the network starts to operates in the Ring-Open state. To provide an alternate path to the broken link path, MRM unblocks one of the ring ports that was blocked earlier in the Ring-Closed state.

The following two figures shows MRM is in a blocked state (as shown by the two parallel lines) because no ports are disconnected, and MRC connections are in the disabled state because the link between them is broken, as marked by a red “x”.

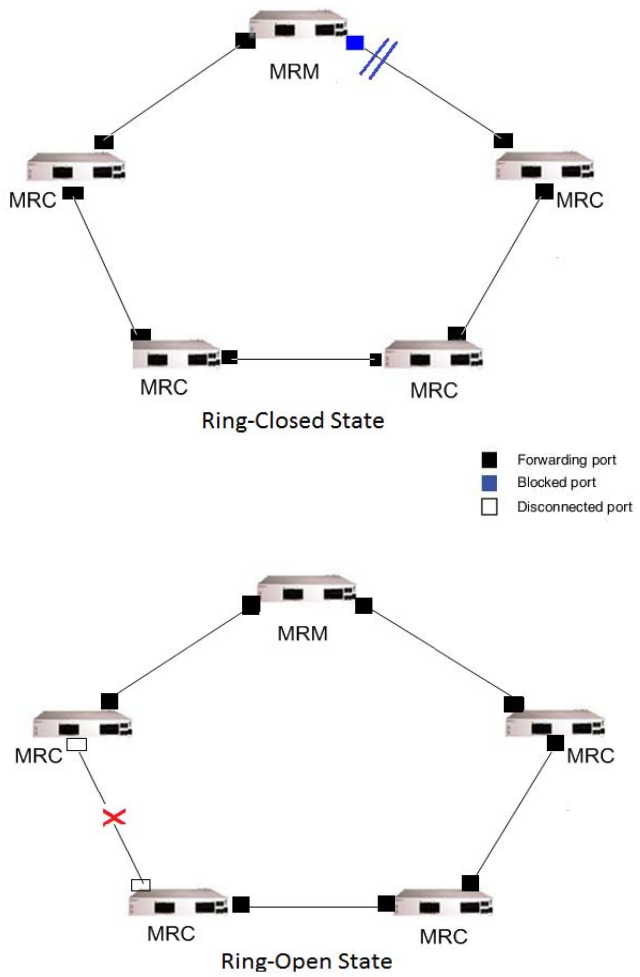


Figure 13-1 : Basic MRP Ring Topology and MRP Ring States

In the case of failure of a link connecting two MRCs, both ring ports of the MRM change to the forwarding state, the MRCs adjacent to the failure have a disabled and a forwarding ring port, and the other MRCs have both ring ports forwarding.

Media Redundancy Manager (MRM)

The function of the MRM is to observe and to control the ring topology in order to react on network faults. One ring port of the MRM shall be connected to a ring port of an MRC. The other ring port of that MRC shall be connected to a ring port of another MRC or to the second ring port of the MRM, thereby forming a ring topology.

The MRM is in constant communication with its MRCs. If one of the links on any of the other ring nodes detects a failure, the client node will send a message to the MRM, which will then unblock its blocked port re-enabling communication between the nodes.

The MRM shall control the ring state by:

- sending MRP_Test frames at a configured time period in both directions of the ring.
- setting one ring port to forwarding state and the other ring port to blocked state if it receives its own MRP_Test frames (this means that the ring is closed).
- setting both ring ports to Forwarding state if it does not receive its own MRP_Test frames on both ring ports within a configured time.

Media Redundancy Client (MRC)

An MRC reacts on received reconfiguration frames from the MRM and can detect and signal link changes on its ring ports. Each MRC shall forward MRP_Test frames received on one ring port to the other ring port and vice versa.

Redundancy Domain

The redundancy domain represents a ring. By default, all MRM and MRCs belong to the default domain. A unique domain ID can be allocated to all the ring nodes of one ring as a key attribute. This is especially essential to distinguish frames if a device is member of multiple rings by implementing multiple instances of MRMs or MRCs. A node shall assign exactly two unique ring ports per redundancy domain.

Media Redundancy Automanager (MRA)

There shall be only one active MRM in the ring even if several nodes have this ability. As an alternative to manual MRP-role configuration, in case of more than one node having the ability to become an MRM in the ring, an enhanced manager voting option may be used to decide which of these nodes shall finally become the MRM, while the other nodes take over the MRC role.

MRM supporting the manager voting option are named MRA. The MRA role is an MRP role, which comprises both the MRP roles MRM and MRC. Additionally, the MRA has the capability to transition at any point in time, by the voting protocol, from its initial operational role manager (MRA) to the operational role client (MRC), and back again, in coordination with possible other MRAs in the ring.

The manager voting process shall be as follows:

- At power on, all MRAs begins the manager voting process. Each MRA begins to send MRP_Test frames on both ring-ports. The MRP_Test frame contains the MRAs priority value. Received MRP_Test frames are forwarded to the local host only.
- The remote manager’s priority values contained in the received MRP_Test frames are compared with their own priority. If their own priority is higher than the received priority, a negative test manager acknowledgment ("MRP_TestMgrNAck") frame is sent along with the remote manager’s MAC address.
- If the receiving MRA receives an acknowledgment ("MRP_TestMgrNAck") with their own MAC address, the receiving MRA initiates the transition into the client (MRC) role.
- The MRP_TestPropagate frame informs other MRA devices in client role about the role change and the new higher priority manager. The clients receiving this frame update their higher priority manager information accordingly. This ensures that clients stay in client role in case of a role change of the monitored higher priority manager.

- If the MRM, which has been elected by the manager voting option, is taken out of the ring or fails to operate, the voting process shall start again and the next priority MRA shall become the managing MRM.
- If an additional MRA is inserted into a ring containing an already elected MRM, then the already elected MRM and the newly inserted MRA will exchange MRP_Test frames and will thereby vote the new remaining MRM among each other.

MRP Configuration Overview and Guidelines

This section describes how to use the OmniSwitch Command Line Interface (CLI) to configure MRP on a switch. Consider the following guidelines when configuring MRP:

- AoS supports installation/removal of AoS MRP package. MRP is packaged into a Debian package which can be installed on the switch. To configure MRP in AoS, it is required to install the MRP package using Package Manager commands. For more information on the Package Manager commands, see "Chassis Management and Monitoring Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide.*
- A maximum of three MRP rings (domain) will be supported at a time in a node. Also, a node can have MRM/MRA role for only one Ring, among allowed three rings.
- The maximum number of nodes in a ring is 50.
- MRP and ERP are mutually exclusive. Hence, one of them can be configured at a time on the switch.
- STP will be disabled on the MRP ring ports once the MRP ring is configured with these ports.
- In closed MRP ring, blocking ports on MRM will go in STP blocking state for all the VLANs configured on this port. VLANs that are not part of ring will also be affected due to this.
- MRP is not supported on mobile ports, mirroring ports, link aggregate member ports, VFL port, UNI port, LPS port, UNP port, DHL port, or Openflow ports.
- In MRP implementation, for some of the events, MRP operates only on Service VLAN's Spanning Tree group making the states as Forwarding or Blocking, which benefits in MRP performance. In STP-FLAT mode, under RSTP/STP protocol, all the VLANs share the same Spanning Tree group making it not possible for MRP to operate on Service VLANS's Spanning Tree group separately. The following must be configured for MRP in FLAT mode.

  1. In FLAT mode, change the STP protocol to MSTP:
  ```
  -> test spantree mode flat
  -> test spantree protocol mstp
  ```

  2. Create a MSTI instance on which only the MRP's Service VLAN must be mapped
  ```
  -> test spantree msti 1
  -> test spantree msti vlan 102
  ```

  (vlan 102 is a MRP Service VLAN in this example)

Configuring an MRP Ring

Use the **mrp** command to configure a domain with VLAN ID for the MRP ring.

```
-> mrp domain 1 vlan 10
```

Configuring MRP Ring Parameters

Use **mrp domain** command to configure the MRP ring parameters like domain index, ring ports, admin-role, priority, react on link change, and recovery.

All devices in a ring which are configured with MRP must be part of the same redundancy domain. The domain is identified by domain identifier or domain name.

For MRM:

```
-> mrp domain 1 name domain-2 port1 linkagg 1 port2 1/1/1 admin-role manager rt-
delay-reset recovery-time 200ms
-> mrp domain 1 manager priority 10 linkchange-react enable test-default inter-
val 50
```

MRC (by default Role is Client):

```
-> mrp domain 1 name domain-2 port1 linkagg 1 port2 1/1/1 admin-role client rt-
delay-reset recovery-time 200ms
```

MRA:

```
-> mrp domain 1 name domain-2 port1 linkagg 1 port2 1/1/1 admin-role auto-
manager rt-delay-reset recovery-time 200ms

-> mrp domain 1 manager priority 10 linkchange-react enable test-default inter-
val 50
```

Enable or disable the administrative state of the MRP ring using the **mrp domain** command. By default, the administrative state is in disable state.

```
-> mrp domain 1 admin-state enable
```

Verifying MRP Configuration

To display information about the MRP configuration on the switch, use the show commands listed below:

show mrp	Displays the MRP configuration information.
show mrp domain	Displays the MRP domain configuration information for a specific ID, domain name, or UUID.
show mrp port	Displays the MRP domain configuration information and counters for a specific port or linkagg.

14 Configuring MVRP

Multiple VLAN Registration Protocol (MVRP) is standards-based Layer 2 network protocol for automatic configuration of VLAN information on switches. It was defined in the 802.1ak amendment to 802.1Q-2005.

MVRP provides a method to share VLAN information and configure the needed VLANs within a layer 2 network. For example, in order to add a switch port to a VLAN, only the end port, or the VLAN-supporting network device connected to the switch port, has to be reconfigured, and all necessary VLAN trunks are dynamically created on the other MVRP-enabled switches. MVRP helps to maintain VLAN configuration dynamically based on current network configurations.

In This Chapter

This chapter describes the MVRP feature and how to configure it through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. This chapter provides an overview of MVRP and includes the following information:

- "Enabling MVRP" on page 14-7
- "Configuring the Maximum Number of VLANs" on page 14-7
- "Configuring MVRP Registration" on page 14-8
- "Configuring the MVRP Applicant Mode" on page 14-9
- "Modifying MVRP Timers" on page 14-10
- "Restricting VLAN Registration" on page 14-11
- "Restricting Static VLAN Registration" on page 14-11
- "Restricting VLAN Advertisement" on page 14-12

MVRP Defaults

The following table lists the defaults for MVRP configuration.

Parameter Description	Command	Default Value/Comments
Enables or disables MVRP globally on a switch.	**mvrp**	disabled
Enables or disables MVRP on specific ports	**mvrp port**	disabled
Maximum number of VLANs	**mvrp maximum-vlan**	256
Registration mode of the port	**mvrp registration**	normal
Applicant mode of the port	**mvrp applicant**	active
Timer value for join timer.	**mvrp timer join**	600 milliseconds
Timer value for leave timer.	**mvrp timer leave**	1800 milliseconds
Timer value for leaveall timer.	**mvrp timer leaveall**	30000 milliseconds
Timer value for periodic timer.	**mvrp timer periodic-timer**	1 second
Restrict dynamic VLAN registration	**mvrp restrict-vlan-registration**	not restricted
Restrict VLAN advertisement	**mvrp restrict-vlan-advertisement**	not restricted
Restrict static VLAN registration	**mvrp static-vlan-restrict**	By default, ports are assigned to the static VLAN based on MVRP PDU processing.

Quick Steps for Configuring MVRP

The following steps provide a quick tutorial on how to configure MVRP. Each step describes a specific operation and provides the CLI command syntax for performing that operation.

1 Create a VLAN using the **vlan** command. For example:

```
-> vlan 5 name "vlan-5"
```

2 Assign a port to the VLAN using the **vlan members** command. For example:

```
-> vlan 5 members port 1/2
```

3 Tag the port with one or more VLANs using the **vlan members** command. For example:

```
-> vlan 5 members port 1/2 tagged
```

4 Enable MVRP globally on the switch by using the **mvrp** command.

```
-> mvrp enable
```

5 Enable MVRP on the port by using the **mvrp port** command. For example, the following command enables MVRP on port 1/2 of the switch:

```
-> mvrp port 1/2 enable
```

6 *Optional:* Restrict a port from becoming a member of the statically created VLAN by using the **mvrp static-vlan-restrict** command. For example, the following command restricts port 1/5 from becoming a member of static VLAN 10:

```
-> mvrp port 1/5 static-vlan-restrict vlan 10
```

Note. To view the global configuration details of the router, enter the **show mvrp configuration** command. The globally configured details are displayed as shown:

```
> show mvrp configuration

MVRP Enabled : yes,
Maximum VLAN Limit : 256
```

To view the MVRP configuration for a specific port, enter the **show mvrp port** command. The configuration data of the particular port is displayed as shown:

```
> show mvrp port 1/2

MVRP Enabled          : no,
Registrar Mode        : normal,
Applicant Mode        : participant,
Join Timer (msec)     : 600,
Leave Timer (msec)    : 1800,
LeaveAll Timer (msec) : 30000,
Periodic Timer (sec)  : 1,
Periodic Tx Status    : disabled
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in this display.

MRP Overview

Multiple Registration Protocol (MRP) was introduced as a replacement for GARP with the IEEE 802.1ak-2007 amendment. The Multiple VLAN Registration Protocol (MVRP) defines a MRP Application that provides the VLAN registration service.

MVRP provides a mechanism for dynamic maintenance of the contents of dynamic VLAN registration Entries for each VLAN, and for propagating the information they contain to other bridges. This information allows MVRP-aware devices to dynamically establish and update their knowledge of the set of VLANs that currently have active members, and through which ports those members can be reached. The main purpose of MVRP is to allow switches to automatically discover some of the VLAN information that would otherwise have to be manually configured.

MVRP Overview

MVRP acts as an MRP application, sending and receiving MVRP information encapsulated in an Ethernet frame on a specific MAC address. MVRP allows both end stations and bridges in a bridged local area network to issue and revoke declarations relating to membership of VLANs. Each MVRP device that receives the declaration in the network creates or updates a dynamic VLAN registration entry in the filtering database to indicate that the VLAN is registered on the reception port.

In this way, MVRP provides a method to share VLAN information within a layer 2 network dynamically, and configure the required VLANs. For example, in order to add a switch port to a VLAN, only the end port, or the VLAN-supporting network device connected to the switch port, must be reconfigured, and all necessary VLAN trunks are dynamically created on the other MVRP-enabled switches. Without using MVRP, either a manual configuration of VLAN trunks or use of a manufacturer specific proprietary method is necessary. MVRP helps to maintain VLAN configuration dynamically based on current network configurations.

How MVRP Works

An MVRP enabled port sends MRPDUs advertising the VLAN enabling another MVRP aware port receiving advertisements over a link to join the advertised VLAN dynamically. All ports of a dynamic VLAN operate as tagged ports for that VLAN.

An MVRP enabled port can forward an advertisement for a VLAN it learned about from other ports on the same switch. However, the forwarding port does not join that VLAN on its own until an advertisement for that VLAN is received on that same port.

The following example illustrates the VLAN advertisements.

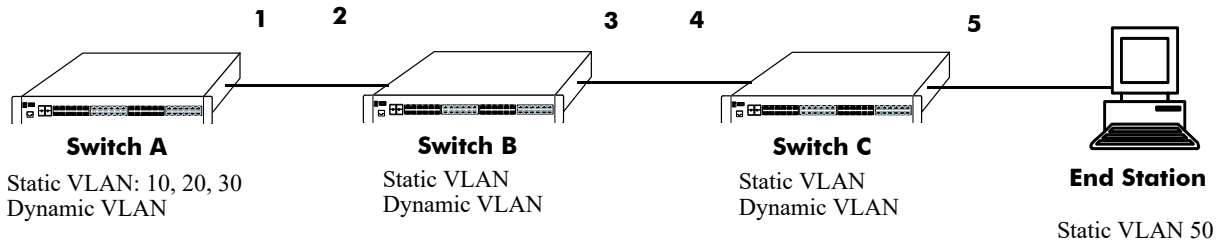


Figure 14-1 : Initial Configuration of MVRP

Switch A has 3 VLANs configured as static VLANs (10, 20, and 30). Other switches on the same network learn these 3 VLANs as dynamic VLANs. Also, the end station connected on port 5 is statically configured for VLAN 50. Port 1 on Switch A is manually configured for VLANs 10, 20, and 30. All the ports are in the same Spanning tree instance and are in forwarding state. Hence, as the diagram shows,

1 Port 1 on Switch A advertises VLAN IDs (VIDs) 10, 20, and 30.

2 Port 2 on Switch B receives the advertisements. VLANs 10, 20, and 30 are created as VLANs on this Switch B and Port 2 become a member of VLANs 10, 20, and 30.

3 Port 3 on Switch B is triggered to advertise VLANs 10, 20, and 30, but does not become a member of these VLANs.

4 Port 4 on Switch C receives the advertisements. VLANs 10, 20, and 30 are created as VLANs on Switch C and Port 4 become a member of VLANs 10, 20, and 30.

5 Port 5 advertises VLANs 10, 20, and 30, but this port is not a member of these VLANs.

Note. Default VLAN (VLAN 1) exists on all switches, but it is not considered here.

The configuration sequence of advertisements and registration of VLANs results in the following configuration.

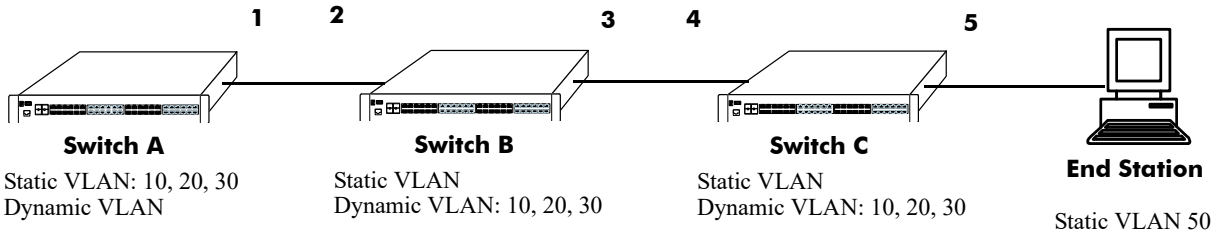


Figure 14-2 : Dynamic Learning of VLANs 10, 20, and 30

Here, the end station advertises itself as a member of VLAN 50. As the **Dynamic Learning of VLANs 10, 20, and 30** diagram shows,

1 Port 5 receives the advertisement and Switch C creates VLAN 50 as a dynamic VLAN. Port 5 of Switch C becomes a member of VLAN 50.

2 Port 4 advertises VLAN 50, but is not a member of VLAN 50.

3 Port 3 of Switch B receives the advertisement, Switch B creates the dynamic VLAN 50, and Port 3 becomes a member of VLAN 50.

4 Port 2 advertises VLAN 50, but is not a member of this VLAN.

5 Port 1 on Switch A receives the advertisement, creates dynamic VLAN 50. Port 1 becomes a member of VLAN 50.

The resulting configuration is depicted as follows:

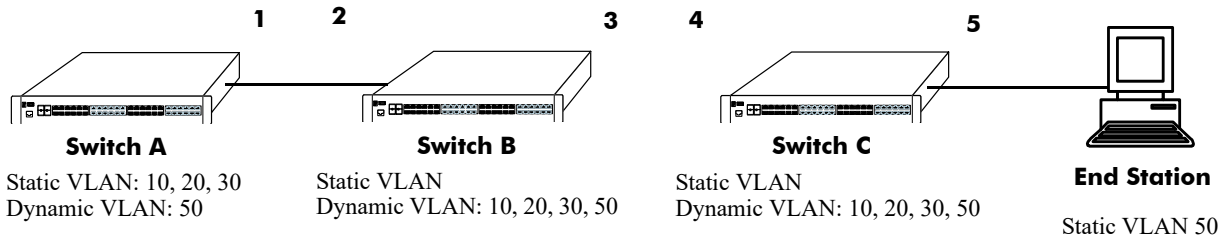


Figure 14-3 : Dynamic Learning of VLAN 50

Note. Every port on a switch is not a member of all the VLANs. Only those ports that receive the advertisement become members of the VLAN being advertised.

Interaction With Other Features

This section contains important information about how other OmniSwitch features interact with MVRP. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

STP

MVRP feature is supported only in STP flat mode. If MVRP is configured in the system with STP flat mode, then STP mode cannot be changed to per-VLAN mode. When a topology change is detected by STP, MAC addresses for the dynamic VPAs learned by MVRP is also deleted.

Configuring MVRP

This section describes how to configure MVRP using the Command Line Interface (CLI) commands.

Enabling MVRP

MVRP is used primarily to prune unnecessary broadcast and unknown unicast traffic, and to create and manage VLANs. MVRP has to be globally enabled on a switch before it can start forwarding MVRP frames. When a port is enabled for MVRP, it cannot be converted as an aggregate or a VLAN stacking User port.

To enable MVRP globally on the switch, enter the **mvrp** command at the CLI prompt as shown:

```
-> mvrp enable
```

To disable MVRP globally on the switch, use disable option of the **mvrp** command as shown:

```
-> mvrp disable
```

Note. Disabling MVRP globally leads to the deletion of all learned VLANs.

MVRP can be enabled on ports regardless of whether it is globally enabled or not. However, for the port to become an active participant, MVRP must be globally enabled on the switch. By default, MVRP is disabled on the ports. To enable MVRP on a specified port, use the **mvrp port** command.

For example, to enable MVRP on port 2 of slot 1, enter:

```
-> mvrp port 1/2 enable
```

Similarly, to enable MVRP on aggregate group 10, enter:

```
-> mvrp linkagg 10 enable
```

To disable MVRP on a specific port, use disable option of the **mvrp port** command as shown:

```
-> mvrp port 1/2 enable
```

Note. MVRP can be configured only on fixed, 802.1 Q and aggregate ports. It cannot be configured on aggregate and VLAN Stacking User ports.

Configuring the Maximum Number of VLANs

A switch can create dynamic VLANs using MVRP. If the VLAN limit to be set is less than the current number of dynamically learned VLANs, then the new configuration takes effect only after the MVRP is disabled and enabled again on the switch. If this operation is not done, the VLANs learned earlier are maintained.

To modify the maximum number of dynamic VLANs the switch is allowed to create, use the **mvrp maximum-vlan** command as shown:

```
-> mvrp maximum-vlan 150
```

Configuring MVRP Registration

MVRP allows a port to register and de-register static VLANs. Every device has a list of all the switches and end stations that can be reached at any given time. When an attribute for a device is registered or de-registered, the set of reachable switches and end stations, also called participants, is modified. Data frames are propagated only to registered devices, thereby preventing attempts to send data to devices that are not reachable.

The following sections describe MVRP registration on switches:

Setting MVRP Normal Registration

The normal registration mode allows dynamic creation, registration, and de-registration of VLANs on a device. The normal mode is the default registration mode.

To configure a port in normal mode, use the **mvrp registration** command. For example, to configure port 2 of slot 1 in normal mode, enter the following:

```
-> mvrp port 1/2 registration normal
```

To view the registration mode of the port, use the **show mvrp port** command. For example:

```
-> show mvrp port 1/2

MVRP Enabled          : no,
Registrar Mode        : normal,
Applicant Mode        : participant,
Join Timer (msec)     : 600,
Leave Timer (msec)    : 1800,

LeaveAll Timer (msec) : 30000,
Periodic Timer (sec)  : 1,
Periodic Tx status    : disabled
```

Setting MVRP Fixed Registration

The fixed registration mode allows only manual registration of the VLANs and prevents dynamic or static de-registration of VLANs on the port.

To configure a port to fixed mode, use the **mvrp registration** command. For example, to configure port 2 of slot 1 to fixed mode, enter the following:

```
-> mvrp port 1/2 registration fixed
```

To view the registration mode of the port, use the **show mvrp port** command. For example,

```
-> show mvrp port 1/2

MVRP Enabled          : no,
Registrar Mode        : fixed,
Applicant Mode        : participant,
Join Timer (msec)     : 600,
Leave Timer (msec)    : 1800,
LeaveAll Timer (msec) : 30000,
Periodic Timer (sec)  : 1,
Periodic Tx status    : disabled
```

Note. The registration mode for the default VLANs of all the ports in the switch is set to normal.

Setting MVRP Forbidden Registration

The forbidden registration mode prevents any VLAN registration or de-registration. If dynamic VLANs previously created are present, they are de-registered.

To configure a port to forbidden mode, use the **mvrp registration** command. For example, to configure port 2 of slot 1 to forbidden mode, enter the following:

```
-> mvrp port 1/2 registration forbidden
```

To view the registration mode of the port, use the **show mvrp port** command. For example,

```
-> show mvrp port 1/2

MVRP Enabled          : no,
Registrar Mode        : forbidden,
Applicant Mode        : participant,
Join Timer (msec)     : 600,
Leave Timer (msec)    : 1800,
LeaveAll Timer (msec) : 30000,
Periodic Timer (sec)  : 1,
Periodic Tx status    : disabled
```

To view the MVRP configurations for all the ports, including timer values, registration and applicant modes, enter the following:

```
-> show mvrp port enable
Port  Join  Leave LeaveAll    Periodic  Registration  Applicant    Periodic
      Timer Timer     Timer    Timer     Mode           Mode          TxStatus
     (msec) (msec)    (msec)   (sec)
----+-------+-------+--------+----------+-------------+------------+--------
1/1   600    1800    30000    2             fixed        active       enabled
1/2   600    1800    30000    2             fixed        active       enabled
1/7   600    1800    30000    2             fixed        active       enabled
1/8   600    1800    30000    2             fixed        active       enabled
2/24  600    1800    30000    2             fixed        active       enabled
```

Configuring the MVRP Applicant Mode

The MVRP applicant mode determines whether MVRP PDU exchanges are allowed on a port, depending on the Spanning Tree state of the port. This mode can be configured to be **participant**, **non-participant**, or **active**. By default, the port is in the **active** mode.

To prevent undesirable Spanning Tree Protocol topology reconfiguration on a port, configure the MVRP applicant mode as active. Ports in the MVRP active applicant state send MVRP VLAN declarations even when they are in the STP blocking state, thereby preventing the STP bridge protocol data units (BPDUs) from being pruned from the other ports.

To set the applicant mode of a port to active, use the **mvrp applicant** command. For example, to set the applicant mode of port 1/2 to active, enter the following:

```
-> mvrp port 1/2 applicant active
```

When a port is set to participant mode, MVRP protocol exchanges are allowed only if the port is set to the STP forwarding state.

To set the applicant mode of port 1/2 to participant mode, enter the following:

```
-> mvrp port 1/2 applicant participant
```

When a port is set to non-participant mode, MVRP PDUs are not sent through the STP forwarding and blocking ports.

To set the applicant mode of port 1/2 to non-participant mode, enter the following:

```
-> mvrp port 1/2 applicant non-participant
```

The applicant mode of the port can be set to the default value by using the **mvrp applicant** command. To set the MVRP applicant mode of port 1/2 to the default mode (active mode), enter the following command:

```
-> mvrp port 1/2 applicant active
```

Modifying MVRP Timers

MVRP timers control the timing of dynamic VLAN membership updates to connected devices. The following are the various timers in MVRP:

- **Join** timer—The maximum time an MVRP instance waits before making declaration for VLANs.
- **Leave** timer—The wait time taken to remove the port from the VLAN after receiving a Leave message on that port.
- **LeaveAll** timer—The time an MVRP instance takes to generate LeaveAll messages. The LeaveAll message instructs the port to modify the MVRP state of all its VLANs to **Leave**.
- **Periodic** timer—The time frequency with which the messages are transmitted again and again.

When you set the timer values, the value for the Leave timer must be greater than or equal to twice the Join timer value plus 100 milliseconds.(**Leave**>=**Join** * 2 +100). The LeaveAll timer value must be greater than or equal to the Leave timer value (**LeaveAll** >= **Leave**). If you attempt to set a timer value that does not adhere to these rules, an error message is displayed.

For example, if you set the Leave timer to 1200 ms and attempt to configure the Join timer to 600 ms, an error is returned. Set the Leave timer to at least 1300 ms and then set the Join timer to 600 ms.

To modify the Join timer value, use the **mvrp timer join** command. For example, to modify the Join timer value of port 1/2, enter the following:

```
-> mvrp port 1/2 timer join 600
```

The Join timer value of port 1/2 is now set to 600 ms.

To set the Leave timer value of port 1/2 to 1800 ms, enter the command as shown:

```
-> mvrp port 1/2 timer leave 1800
```

To set the LeaveAll timer of port 1/2 to 30000 ms, enter the command as shown:

```
-> mvrp port 1/2 timer leaveall 30000
```

To set the Periodic timer of port 1/2 to 1 second, enter the command as shown:

```
-> mvrp port 1/2 timer periodic-timer 1
```

To view the timer value assigned to a particular port, use the **show mvrp timer** command.

```
-> show mvrp port 1/2 timer
Join Timer (msec)       : 600,
Leave Timer (msec)      : 1800,
LeaveAll Timer (msec)   : 30000,
Periodic-Timer (sec)    : 1
```

Note. Set the same MVRP timer value on all the connected devices.

Restricting VLAN Registration

Restricted VLAN registration restricts MVRP from dynamically registering specific VLAN or VLANs on a switch. It decides whether VLANs can be dynamically created on a device or only be mapped to the ports (if the VLANs are already statically created on the device).

By default, the dynamic VLAN registrations are not restricted and the VLAN can either be created on the device or mapped to another port.

To restrict a VLAN from being dynamically learned on the device, you can configure the dynamic VLAN registrations by using the **mvrp restrict-vlan-registration** command as shown:

```
-> mvrp port 1/1 restrict-vlan-registration vlan 4
```

Here, VLAN 4 cannot be learned by the device dynamically. However, if the VLAN exists on the device as a static VLAN, it can be mapped to the receiving port.

To allow dynamic VLAN registrations on the port, use the no form of the **mvrp restrict-vlan-registration** command as shown:

```
-> no mvrp port 1/1 restrict-vlan-registration vlan 4
```

Restricting Static VLAN Registration

Ports can be exempted from becoming members of statically created VLANs. To restrict a port from becoming a member of a statically configured VLAN, use the **mvrp static-vlan-restrict** command as shown:

```
-> mvrp port 1/9 static-vlan-restrict vlan 5
```

Note. This command does not apply to dynamic VLANs.

Here, the port 1/9 is restricted from becoming a MVRP member of VLAN 5.

To restrict a port from becoming a member of a range of statically created VLANs, enter the **mvrp static-vlan-restrict** command as shown:

```
-> mvrp port 1/9 static-vlan-restrict vlan 5-9
```

Here, port 1/9 is restricted from becoming a MVRP member of VLANs 5 to 9.

A port can be allowed to become a member of statically created VLANs using the no form of the **mvrp static-vlan-restrict** command. To allow port 1/2 to become a member of a statically created VLAN, enter the command as shown:

```
-> no mvrp port 1/2 static-vlan-restrict vlan 5-9
```

Restricting VLAN Advertisement

VLANs learned by a switch through MVRP can either be propagated to other switches or be blocked. This helps prune VLANs that have no members on a switch. If the applicant mode is set to participant or active, you can use the **mvrp restrict-vlan-advertisement** command to restrict the propagation of VLAN information on a specified port as shown:

```
-> mvrp port 1/1 restrict-vlan-advertisement vlan 5
```

Here, VLAN 5 is not allowed to propagate on port 1 of slot 1.

To enable the propagation of dynamic VLANs on the specified port, use the no form of the command. To restrict VLAN 5 from being propagated to port 1/1, enter the command as shown:

```
-> no mvrp port 1/1 restrict-vlan-advertisement vlan 5
```

Verifying the MVRP Configuration

A summary of the commands used for verifying the MVRP configuration is given here:

show mvrp last-pdu-origin	Displays the source MAC address of the last MVRP message received on specific ports or aggregates.
show mvrp configuration	Displays the global configuration for MVRP.
show mvrp linkagg	Displays the MVRP configuration for a specific port or an aggregate of ports.
show mvrp port	Displays the MVRP configurations for all the ports, including timer values, registration and applicant modes.
show mvrp vlan-restrictions	Displays the list of VLANS learned through MVRP and their details.
show mvrp timer	Displays the timer values configured for all the ports or a specific port.
show mvrp statistics	Displays the MVRP statistics for all the ports, aggregates, or specific ports.
clear mvrp statistics	Clears MVRP statistics for all the ports, an aggregate of ports, or a specific port.

For more information about the output details that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

15 Configuring 802.1AB

Link Layer Discovery Protocol (LLDP) is an emerging standard that provides a solution for the configuration issues caused by expanding networks. LLDP supports the network management software used for complete network management. LLDP is implemented as per the IEEE 802.1AB standard. LLDP specifically defines a standard method for Ethernet network devices and Media Endpoint Devices (MED) to exchange information with its neighboring devices and maintain a database of the information. The exchanged information, passed as LLDPDU, is in TLV (Type, Length, Value) format. The information available to the network management software must be as new as possible; hence, remote device information is periodically updated.

In This Chapter

This chapter describes the basic components of 802.1AB and how to configure them through the Command Line Interface (CLI). The CLI commands are used in the configuration examples; for more details about the syntax of commands, see Chapter 17, "802.1AB Commands," in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include the following:

- "Quick Steps for Configuring 802.1AB" on page 15-3
- "802.1AB Overview" on page 15-4.
- "Configuring LLDPDU Flow" on page 15-8.
- "Enabling and Disabling Notification" on page 15-8.
- "Enabling and Disabling Management TLV" on page 15-9.
- "Enabling and Disabling 802.1 TLV" on page 15-9.
- "Enabling and Disabling 802.3 TLV" on page 15-9.
- "Enabling and Disabling MED TLV" on page 15-10.
- "Enabling and Disabling Proprietary TLV" on page 15-10
- "Enabling and Disabling Application Priority TLV" on page 15-12.
- "Setting the Transmit Interval" on page 15-13.
- "Setting the Transmit Hold Multiplier Value" on page 15-13.
- "Setting the Reinit Delay" on page 15-13.
- "Setting the Notification Interval" on page 15-13.
- "Verifying 802.1AB Configuration" on page 15-15.

802.1AB Defaults Table

The following table shows the default settings of the configurable 802.1AB parameters.

Parameter Description	Command	Default Value/Comments
Transmit time interval for LLDPDUs	**lldp transmit interval**	30 seconds
Transmit hold multiplier value	**lldp transmit hold-multiplier**	4
Reinit delay	**lldp reinit delay**	2 seconds
Notification interval	**lldp notification interval**	5 seconds
LLDPDUs transmission	**lldp lldpdu**	Transmission and Reception
Per port notification	**lldp notification**	Disable
Management TLV	**lldp tlv management**	Disable
802.1 TLV	**lldp tlv dot1**	Disable
802.3 TLV	**lldp tlv dot3**	Disable
LLDP Media Endpoint Device	**lldp tlv med**	Disable

Quick Steps for Configuring 802.1AB

1 To enable the transmission and the reception of LLDPDUs on a port, use the **lldp lldpdu** command. For example:

```
-> lldp port 2/47 lldpdu tx-and-rx
```

2 To control per port notification status about a change in a remote device associated to a port, use the **lldp notification** command. For example:

```
-> lldp port 2/47 notification enable
```

3 To control per port management TLV to be incorporated in the LLDPDUs, use the **lldp tlv management** command. For example:

```
-> lldp port 2/47 tlv management port-description enable
```

4 Set the transmit time interval for LLDPDUs. To set the timer for a 50 second delay, use the **lldp transmit interval** command. For example:

```
-> lldp transmit interval 50
```

5 Set the LLDPDUs transmit fast start count required for LLDP Fast Restart mechanism to be activated.

Note. *Optional*. Verify the LLDP per port statistics by entering the **show lldp statistics** command. For example:

```
-> show lldp statistics
           LLDPDU    LLDPDU   LLDPDU    LLDPDU   LLDPDU     TLV      TLV
    Device
Slot/Port    Tx     TxLenErr    Rx      Errors   Discards  Unknown  Discards
Ageouts
---------+---------+---------+---------+---------+---------+---------+---------+----
----
   1/1        453        0       452         0         0         0         0       0
   1/2        452        0       453         0         0         0         0       0
   1/5        452        0       473         0         0       476       476       0
   1/8        455        0       464         0         0         0         0       0
   1/9        456        0       464         0         0         0         0       0
```

To verify the remote system information, use the **show lldp remote-system** command. For example:

```
-> show lldp remote-system
Remote LLDP Agents on Local Slot/Port: 2/47,
  Chassis ID Subtype          = 4 (MAC Address),
  Chassis ID                  = 00:d0:95:e9:c9:2e,
  Port ID Subtype             = 7 (Locally assigned),
  Port ID                     = 2048,
  Port Description            = (null),
  System Name                 = (null),
  System Description          = (null),
  Capabilites Supported       = none supported,
  Capabilites Enabled         = none enabled,
```

For more information about this display, see the *OmniSwitch AOS Release 8 CLI Reference Guide.*

802.1AB Overview

LLDP is a Layer 2 protocol used to detect adjacent devices in a network. Each device in a network sends and receives LLDPDUs through all ports on which the protocol is enabled. If the protocol is disabled on a port, then LLDPDUs received on that port are dropped.

The LLDPDUs are transmitted at a certain interval. This transmission interval can be configured. When an LLDPDU is received from a neighboring device, the LLDPDU software validates the frame and stores the information in the remote device Management Information Base (MIB). This information ages periodically. If an LLDPDU is not received from the same device within the time specified in the TTL TLV of the LLDPDU, the information is updated in the related MIB. By exchanging information with all the neighbors, each device gets to know its neighbor on each port. The information contained in the LLDPDU is transmitted in the TLV (Type, Length, Value) format and falls under two categories:

- Mandatory
- Optional

Each LLDPDU contains all the five mandatory TLVs and optional TLVs.

Mandatory TLVs

The mandatory TLV information contains the following information with regard to the LAN device:

- MSAP (MAC service access point) identifier.
- Time period for the validity of the information

The mandatory TLVs contained in an LLDPDU are listed below:

- Chassis ID TLV
- Port ID TLV
- VLAN ID TLV
- Time to live TLV
- End of LLDPDU TLV

Optional TLVs

The optional TLVs defined as part of LLDP are grouped into the following sets listed below:

Basic Management TLV Set

- Port Description TLV
- System Name TLV
- System Description TLV
- System capabilities TLV
- Management address TLV

Note. This optional TLV set is required for all LLDP implementation.

IEEE 802.1 Organizationally Specific TLV Set

- Port VLAN ID TLV
- Port and Protocol VLAN ID TLV
- VLAN name TLV
- Protocol identity TLV

Note. If one TLV from this set is included in the LLDPDU, then all the other TLVs need to be included.

IEEE 802.3 Organizationally Specific TLV Set

- MAC/PHY configuration/status TLV
- Power via MDI TLV (in network connectivity TLV set, Extended Power-through-MDI TLV).
- Power via MDI extension (29 Bytes) TLV to support type 3 and type 4 PD for IEEE802.3bt.
- Power via MDI measurements (28 byte) TLV to support IEEE802.3bt.
- Link Aggregation TLV
- Maximum frame size TLV

ANSI-TIA LLDP-MED TLV Sets

- Network connectivity TLV set
- LLDP-MED capabilities TLV
- Inventory Management TLV
- Location Identification TLV
- Extended Power-through-MDI TLV

When an 802.1AB supporting system receives an LLDPDU containing MED capability TLV, then the remote device is identified as an edge device , for example, IP phone and IP PBX, among others. In such a case, the switch stops sending LLDPDU and starts sending MED LLDPDU on the port connected to the edge device.

LLDP-Media Endpoint Devices

LLDP-MED is an extension to 802.1ab (Link Layer Discovery Protocol - LLDP), a link-layer protocol that defines a method for network access devices using Ethernet connectivity to advertise device information, device capabilities and media specific configuration information periodically to peer devices attached to the same network.

The LLDP-MED feature facilitates the information sharing between Media Endpoint Devices and Network Infrastructure Devices. It is designed to allow the following functionalities:

- Auto-discovery of LAN policies (such as VLAN, Layer 2 Priority and Diffserv settings) leading to "plug and play" networking. This is achieved by advertising the VLAN information.
- Device location discovery to allow creation of location databases for VoIP, E911 services.

- Extended and automated power management of Power-over-Ethernet endpoints.
- Inventory management, allowing network administrators to track their network devices, and determine their characteristics (manufacturer, software and hardware versions, and serial / asset number).
- Support for receiving, storing and advertising of VLAN information from and to remote Network Connectivity Devices and Media Endpoint Devices (MEDs).
- Support for receiving and storing of Inventory Management TLVs from remote Media Endpoint Devices.

LLDP Agent Operation

A network device that implements LLDP, supports an LLDP agent. An LLDP agent operates in any one of the following three modes:

Transmit-only mode: The agent can only transmit the information about the capabilities and the current status of the local system at regular intervals.

Receive-only mode: The agent can only receive information about the capabilities and the current status of the remote systems.

Transmit and receive mode: The agent can transmit the capabilities and status information of the local system and receive the capabilities and the status information of the remote system.

LLDPDU Transmission and Reception

LLDP operates in a one-way direction, so that the information in the LLDPDUs flows from one device to another. LLDPDUs are not exchanged as an information request by one device and a response sent by another device. The other devices do not acknowledge LLDP information received from a device.

The transmission of LLDPDU is based on two factors:

- Transmit countdown timing counter. For example, whenever the counter expires, it goes through the entire database of ports that have links and sends the LLDPDU when the current time has exceeded the re-transmission time interval.
- If there is change in status of any of the ports. For example, a new port is attached or a new link has come up.

Reception of LLDPDU is a two-phase process:

- LLDPDU and TLV error handling as per the 802.1AB standard
- LLDP remote system MIB update

Aging Time

The LLDP specific information of the remote system is stored in the LLDP MIB. The TTL TLV carries a positive value in seconds, and conveys to the other device the duration for which this information is valid. Once a remote device is learned on a local port, if the receiving device does not receive an LLDPDU from the same remote device and on the same local port within the TTL mentioned in the previous LLDPDU, then the local device discards the related entry from its database. This is called the aging time and can be set by the user.

LLDP Agent Security Mechanism

The OmniSwitch LLDP Agent Security mechanism provides a solution for secure access to the network by detecting rogue devices and preventing them from accessing the internal network. LLDP agent security can be achieved by allowing only one trusted LLDP remote agent on a network port.

User is provided an option to configure the Chassis ID subtype that can be used in validating the Chassis ID type in the incoming LLDP PDU. If the Chassis ID is not configured, by default, the first LLDP remote agent is learnt with the received Chassis ID. When more than one LLDP agent is learned on a port, the port is moved to a violation state.

For example, when someone tries to take control over the network by connecting non-registered devices to an NNI port, the LLDP Security mechanism is activated. One or both of the following actions are performed according to the security configuration:

- When the rogue device is detected, a violation is reported on the port.
- The NNI port that is connected to the rogue device is blocked. Thus the rogue device is prevented from accessing the internal network.

LLDP security mechanism can be enabled or disabled globally at chassis level, at slot level, or at individual port level. When the LLDP agent security is enabled, the configured ports are monitored for reception of any LLDPDU. When an LLDPDU is received, the remote agent ID is learned and the port is considered as a trusted port if the port does not have any other LLDP remote agent assigned. If the remote agent chassis ID and port IDs received are already present in the trusted remote agent database on the same port, then the port remains in a trusted state.

However, a port is moved to violation state under the following conditions:

- When a link up is received on a LLDP security enabled port, if no LLDPDU is received even after three times the LLDP timer interval period (30 seconds), the port is moved to a violation state.
- If a trusted remote agent exists, and if no LLDP remote agent is learned even after three times the LLDP timer interval period (30 seconds), the port is moved to a violation state.
- If a new LLDP remote agent is learned after the link up and down, then the port is moved to a violation state.
- If the same chassis ID and port ID exist in the trusted remote agent database but on a different port, then the port remote agent is learned and the port is moved to a violation state.
- If a new LLDP remote agent is learned on a port that has a trusted LLDP remote agent, then the port is moved to a violation state.

Three actions can be configured when an LLDP security violation occurs. The different violation actions that can be configured are:

- **trap** - Generate a trap
- **shutdown** - Shutdown the port
- **trap-and-shutdown** - A trap is generated upon shutdown of the port due to violation.

When a shutdown occurs on a port, it can be cleared manually through the CLI interface using the **clear violations** command.

Configuring 802.1AB

The following sections list detail procedures to enable 802.1AB and assign ports to 802.1AB.

Configuring LLDPDU Flow

The **lldp lldpdu** command can be used to enable or disable the LLDPDU flow on a specific port, a slot, or all ports on a switch. When enabled, the port can be set to receive, transmit, or to transmit and receive LLDPDUs.

To set the LLDPDU flow on a switch as transmit and receive, enter the **lldp lldpdu** command:

```
-> lldp chassis lldpdu tx-and-rx
```

To set the LLDPDU flow on port 4 of slot 3 as receive, enter the following command at the CLI prompt:

```
-> lldp 3/4 lldpdu rx
```

To disable the flow of LLDPDU on a switch, enter the **lldp lldpdu** command:

```
-> lldp chassis lldpdu disable
```

To disable the flow of LLDPDU on port 5 of slot 1, enter the following command at the CLI prompt:

```
-> lldp 1/5 lldpdu disable
```

Enabling and Disabling Notification

The **lldp notification** command is used to control per port notification status about the remote device change on a specific port, a slot, or all ports on a switch. When enabled, the LLDPDU administrative status must be in the receive state.

To enable notification of local system MIB changes on a switch, enter the **lldp notification** command:

```
-> lldp chassis notification enable
```

To enable notification on port 2 of slot 1, enter the following command at the CLI prompt:

```
-> lldp port 1/2 notification enable
```

To disable notification on a switch, enter the **lldp notification** command:

```
-> lldp chassis notification disable
```

To disable notification on port 4 of slot 1, enter the following command at the CLI prompt:

```
-> lldp port 1/4 notification disable
```

Enabling and Disabling Management TLV

The **lldp tlv management** command is used to control per port management TLVs transmission in the LLDPDUs on a specific port, a slot, or all ports on a switch. When enabled, the LLDPDU administrative status must be in the transmit state.

To enable the management TLV LLDPDU transmission on a switch, enter the **lldp tlv management** command:

```
-> lldp chassis tlv management port-description enable
```

To enable the management TLV on port 3 of slot 2, enter the following command at the CLI prompt:

```
-> lldp port 2/3 tlv management system-capabilities enable
```

To disable the management TLV on a switch, enter the **lldp tlv management** command:

```
-> lldp chassis tlv management port-description disable
```

To disable management TLV on port 3 of slot 2, enter the following command at the CLI prompt:

```
-> lldp port 2/3 tlv management system-capabilities disable
```

Enabling and Disabling 802.1 TLV

The **lldp tlv dot1** command is used to control per port 802.1 TLVs transmission in the LLDPDUs on a specific port, a slot, or all ports on a switch. When enabled, the LLDPDU administrative status must be in the transmit state.

To enable the 802.1 TLV LLDPDU transmission on a switch, enter the **lldp tlv dot1** command:

```
-> lldp chassis tlv dot1 port-vlan enable
```

To enable the 802.1 TLV on port 1 of slot 5, enter the following command at the CLI prompt:

```
-> lldp port 5/1 tlv dot1 vlan-name enable
```

To disable the 802.1 TLV on a switch, enter the **lldp tlv dot1** command:

```
-> lldp chassis tlv dot1 port-vlan disable
```

To disable 802.1 TLV on port 2 of slot 5, enter the following command at the CLI prompt:

```
-> lldp port 5/2 tlv dot1 vlan-name disable
```

Enabling and Disabling 802.3 TLV

The **lldp tlv dot3** command is used to control per port 802.3 TLVs transmission in the LLDPDUs on a specific port, a slot, or all ports on a switch. When enabled, the LLDPDU administrative status must be in the transmit state.

To enable the 802.3 TLV LLDPDU transmission on a switch, enter the **lldp tlv dot3** command, as shown:

```
-> lldp chassis tlv dot3 mac-phy enable
```

To enable the 802.3 TLV on port 4 of slot 2, enter the following command at the CLI prompt:

```
-> lldp port 2/4 tlv dot3 mac-phy enable
```

To disable the 802.3 TLV on a switch, enter the **lldp tlv dot3** command, as shown:

```
-> lldp chassis tlv dot3 mac-phy disable
```

To disable 802.3 TLV on port 5 of slot 3, enter the following command at the CLI prompt:

```
-> lldp port 3/5 tlv dot3 mac-phy disable
```

Enabling and Disabling MED TLV

The **lldp tlv med** command is used to control per port LLDP Media End Device (MED) TLVs transmission in the LLDPDUs on a specific port, a slot, or all ports on a switch. When enabled, the LLDPDU administrative status must be in the transmit state.

To enable the LLDP-MED TLV LLDPDU transmission on a switch, enter the **lldp tlv med** command, as shown:

```
-> lldp chassis tlv med power enable
```

To enable the MED TLV on port 4 of slot 4, enter the following command at the CLI prompt:

```
-> lldp port 4/4 tlv med capability enable
```

To disable the MED TLV on a switch, enter the **lldp tlv med** command, as shown:

```
-> lldp chassis tlv med power disable
```

To disable MED TLV on port 3 of slot 4, enter the following command at the CLI prompt:

```
-> lldp port 4/3 tlv med capability disable
```

Enabling and Disabling Proprietary TLV

The OmniSwitch advertise the Access Point location information to the OmniAccess Stellar APs connected to it through the Proprietary TLVs. The proprietary TLVs are transmitted along with the LLDP BPDU. After LLDP is configured on the network devices, the NMS can obtain the network topology.

The WLAN management VLAN is transmitted to OmniAccess Stellar AP through LLDP using existing Port VLAN TLV, even if the TLV is disabled by LLDP module.

The Proprietary TLV must be enabled to advertise the OmniAccess Stellar AP location. The **lldp tlv proprietary** command is used to enable the Proprietary TLV.

Note. The OmniAccess Stellar AP should always be connected to the UNP Port.

To enable the Proprietary TLV transmission on a switch, enter the **lldp tlv proprietary** command, as shown:

```
-> lldp chassis tlv proprietary enable
```

To enable the Proprietary TLV on port, for example port 4 of slot 4, enter the following command at the CLI prompt:

```
-> lldp port 1/4/4 tlv proprietary enable
```

To disable the Proprietary TLV on a switch, enter the **ldp tlv proprietary** command, as shown:

```
-> lldp chassis tlv proprietary disable
```

To disable Proprietary TLV on port, for example port 3 of slot 4, enter the following command at the CLI prompt:

```
-> lldp port 1/4/4 tlv proprietary disable
```

To view the operational status of Proprietary TLV on a switch, use the **show lldp config** command.

```
-> show lldp config
----------+-------------------------------------------+--------------------+---------+--------------
          |  Admin    |  Notify   |  Std TLV  |   Mgmt    |  802.1    |  802.3    |   MED   |  Proprietary
 Slot/Port|  Status   |   Trap    |   Mask    | Address   |   TLV     |   Mask    |   Mask  |     TLV
----------+-----------+-----------+-----------+-----------+-----------+-----------+---------+--------------
   1/1      Rx + Tx    Disabled     0x00      Disabled   Disabled     0x00       0x00      Disabled
   1/2      Rx + Tx    Disabled     0x00      Disabled   Disabled     0x00       0x00      Disabled
   1/3      Rx + Tx    Disabled     0x00      Disabled   Disabled     0x00       0x00      Disabled
   1/4      Rx + Tx    Disabled     0x00      Disabled   Disabled     0x00       0x00      Disabled
   1/5      Rx + Tx    Disabled     0x00      Disabled   Disabled     0x00       0x00      Disabled
   1/6      Rx + Tx    Disabled     0x00      Disabled   Disabled     0x00       0x00      Disabled
   1/7      Rx + Tx    Disabled     0x00      Disabled   Disabled     0x00       0x00      Disabled
   1/8      Rx + Tx    Disabled     0x00      Disabled   Disabled     0x00       0x00      Disabled
   1/9      Rx + Tx    Disabled     0x00      Disabled   Disabled     0x00       0x00      Disabled
   1/10     Rx + Tx    Disabled     0x00      Disabled   Disabled     0x00       0x00      Disabled
```

To view the AP location, use the **show lldp local-port** command.

```
-> show lldp local-port

Local Slot 1/Port 1 LLDP Info:
  Port ID                     = 1001 (Locally assigned),
  Port Description            = Alcatel-Lucent 1/1,
  Vlan                        = 1,
  AP Location                 = sw1,
Local Slot 1/Port 2 LLDP Info:
  Port ID                     = 1002 (Locally assigned),
  Port Description            = Alcatel-Lucent 1/2,
  Vlan                        = 1,
  AP Location                 = -,
```

The AP location and VLAN information can also be viewed from WebView. To view the information, from the WebView page:

1 Click on the **Physical** tab.

2 Click on **Adjacencies**. Adjacencies home page is displayed.

3 In the Adjacencies home page, select **LLDP**.

4 Click on **Local** and select **AP Management TLV** from the displayed option. This will display the **AP Management TLV** information such as Slot, Port, VLAN, and AP Location. A sample screen is displayed as follows:

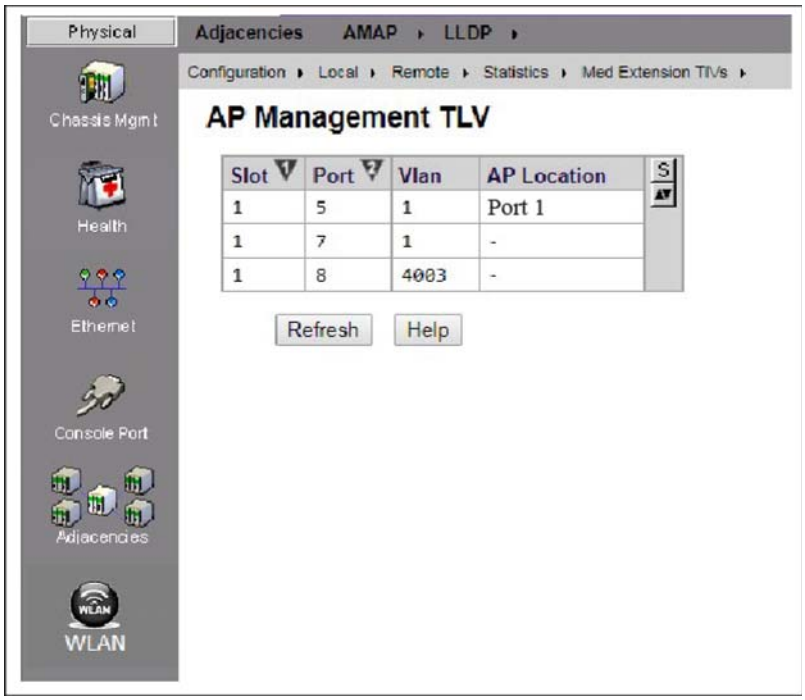


Figure 15-1 : Enabling and Disabling Proprietary TLV

Note. For more information about WebView, see the *OmniSwitch AOS Release 8 Switch Management Guide*.

Enabling and Disabling Application Priority TLV

The **lldp tlv application** command is used to include the LLDP-DCBx Application Priority TLV in the LLDPDUs transmitted on a specific port, a slot, or all ports on a switch. When enabled, the LLDPDU administrative status must be in the transmit state.

To enable the Application Priority TLV LLDPDU transmission on a switch, enter the **lldp tlv application** command, as shown:

```
-> lldp chassis tlv application enable
```

To enable the Application Priority TLV on port 4 of slot 4, enter the following command at the CLI prompt:

```
-> lldp port 4/4 tlv application enable
```

To disable the Application Priority TLV on a switch, enter the **lldp tlv application** command, as shown:

```
-> lldp chassis tlv application disable
```

To disable the Application Priority TLV on port 3 of slot 4, enter the following command at the CLI prompt:

```
-> lldp port 4/3 tlv application disable
```

Configuring Application Priority TLV Parameters

The **lldp tlv application priority** command is used to configure the the LLDP-DCBx Application Priority TLV to advertise an 802.1p priority value for specific protocols on a specific port, a slot, or all ports on a switch. The LLDPDU administrative status must be enabled and set to transmit and receive before using this command. In addition, the Application Priority TLV must be enabled for transmission in the LLDPDU.

To configure Application Priority TLV parameters, enter the **lldp tlv application** command, as shown:

```
-> lldp port 1/1/3 tlv application fcoe priority 3
-> lldp port 1/1/3 tlv application tcp-sctp-port 3192 priority 5
```

To remove Application Priority TLV parameters, use the **no** form of the **lldp tlv application** command, as shown:

```
-> lldp chassis tlv application no fcoe
-> lldp chassis tlv application no tcp-sctp-port
```

Setting the Transmit Interval

To set the transmit time interval for LLDPDUs, enter the **lldp transmit interval** command. For example, to set the transmit time interval as 40 seconds, enter:

```
-> lldp transmit interval 40
```

Setting the Transmit Hold Multiplier Value

To set the transmit hold multiplier value, enter the **lldp transmit hold-multiplier** command. For example, to set the transmit hold multiplier value to 2, enter:

```
-> lldp transmit hold-multiplier 2
```

Note. The Time To Live is a multiple of the transmit interval and transmit hold-multiplier.

Setting the Reinit Delay

To set the time interval that must elapse before the current status of a port is reinitialized after a status change, enter the **lldp reinit delay** command. For example, to set the reinit delay to 7 seconds, enter:

```
-> lldp reinit delay 7
```

Setting the Notification Interval

To set the time interval that must elapse before a notification about the local system Management Information Base (MIB) change is generated, enter the **lldp notification interval** command. For example, to set the notification value to 130 seconds, enter:

```
-> lldp notification interval 130
```

Note. In a specified interval, generating more than one notification-event is not possible.

Application Example - LLDP MED

The following example describes how to configure LLDP MED on the devices.

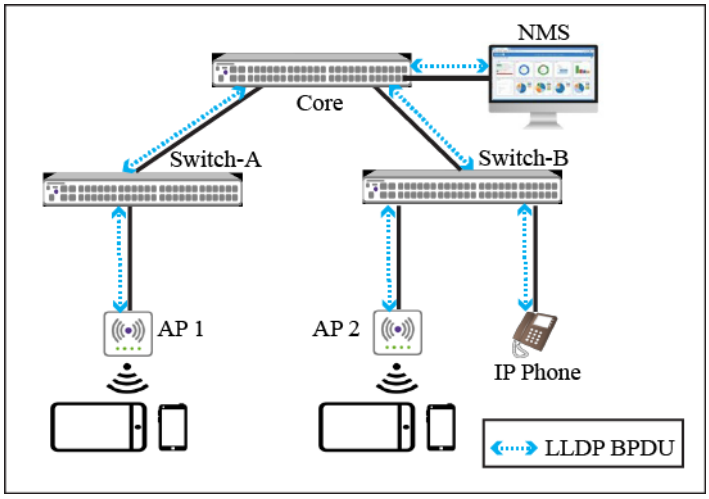


Figure 15-2 : Application Example - LLDP MED

In the above example, the NMS obtains Layer 2 information about Core Switch, SwitchA, SwitchB, and AP. By using the Layer 2 information, a network administrator can know the detailed network topology information and configuration conflicts. These requirements can be met by configuring LLDP on SwitchA and SwitchB. In addition, the administrator requires that SwitchA and SwitchB send LLDP traps to the NMS, when the LLDP management address changes, global LLDP is enabled or disabled.

For more information on the configuration procedure, see "Configuring 802.1AB" on page 15-8

Verifying 802.1AB Configuration

To display information about the ports configured to handle 802.1AB, use the following show command:

show lldp system-statistics	Displays system-wide statistics.
show lldp statistics	Displays port statistics.
show lldp local-system	Displays local system information.
show lldp local-port	Displays port information.
show lldp local-management-address	Displays the local management address information.
show lldp config	Displays the general LLDP configuration information for LLDP ports.
show lldp remote-system	Displays local port information of remote system.
show lldp remote-system med	Displays MED local port information of remote system.
show lldp remote-system application-tlv	Displays Application Priority TLV information of the remote system.

For more information about the resulting display, see Chapter 17, "802.1AB Commands," in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

16 Configuring SIP Snooping

Session Initiation Protocol (SIP) address the key challenge of real time delivery and monitoring requirements for media streams from SIP devices.

SIP Snooping prioritizes voice and video traffic over non-voice traffic.

- Identifies and marks the SIP and its corresponding media streams. Each media stream contains Real Time Protocol (RTP) and Real Time Control Protocol (RTCP) flows. Marking is done using the DSCP field in the IP header.
- Provides user configured QOS treatment for SIP/RTP/RTCP traffic flows based on its marking.
- Also snoops voice quality metrics of media streams from their RTCP packets and displays them to the user with knowledge of media reception quality in real time and helps to diagnose the problems on their quality. Also in addition, trap will be generated when voice quality parameters like Jitter, Round trip time, Packet-lost, R-factor and MOS values of media streams crosses user configured threshold.

In This Chapter

This chapter describes the SIP Snooping feature, and how to configure it through the Command Line Interface (CLI). For more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following information and procedures are included in this chapter:

- "Quick Steps for Configuring SIP Snooping" on page 16-4.
- "SIP Snooping Overview" on page 16-5
- "SIP Snooping Configuration Guidelines" on page 16-8
- "SIP Snooping Limitations" on page 16-15
- "Verifying the SIP Snooping Configuration" on page 16-16.

SIP Snooping Defaults

The following table shows SIP Snooping default values.

Parameter Description	Command	Default Value/Comments
The administrative status of SIP Snooping	**sip-snooping admin-state**	disable
Configure the status of SIP snooping	**sip-snooping port admin-state**	disable
SIP Snooping mode	**sip-snooping mode**	automatic
Configure IP address of the trusted servers	**sip-snooping trusted server**	none
Configure SIP PDU DSCP marking configuration.	**sip-snooping sip-control**	By default, DSCP is not marked on the switch.
Configure the SOS call strings	**sip-snooping sos-call number**	none
Configure the SOS-Call RTP/RTCP DSCP Marking	**sip-snooping sos-call dscp**	EF/46
Configure the UDP port of the switch	**sip-snooping udp port**	none
Configure the TCP port of the switch	**sip-snooping tcp port**	port 5260

Parameter Description and Values

No	PARAMETER Description	Default value	Configurable	Min	Max
1	Global SIP snooping	Disable	YES	NA	NA
2	SIP snooping per port	Enable	YES	NA	NA
3	SIP Snooping mode	Automatic	YES	NA	NA
4	Number of SIP UDP Ports	NO	YES	0	8
5	Number of SIP TCP Ports	5260	YES	0	8
6	Number of Trusted Call server	NO	YES		8
7	Number of SOS-Call	NO	YES	0	4
8	SOS-Call RTP/RTCP Bandwidth	128 kbps	NO	NA	NA
9	SOS-Call RTP/RTCP-DSCP	46 EF	YES	NA	NA
10	SIP control DSCP	NO	YES	NA	NA
11	SIP rate limit	1 mbps	NO	1	4
12	Media Idle timeout	5 min	NO	NA	NA
13	RTCP monitoring	Enable	YES	NA	NA
14	Jitter Threshold (audio/video/other)	50/100/100 ms	YES	0	300
15	Packet-lost Threshold (audio/video/other)	10 /20/20 %	YES	0	99
16	RTT Threshold (audio/video/other)	180 /250/250 ms	YES	0	500
17	R-factor Threshold (audio/video/other)	70/80/80	YES	0	100
18	MOS Threshold (audio/video/other)	3.6/3.0/3.0	YES	0	5
19	TCAM slice reserved	1	NO	1	4
20	Number of Media streams per NI[a]	(64*TCAM Slice value) – 4	NO	NA	NA
21	Number of Media streams per system in case of stack. (VC with MAX_NI_SLOTS = 8)	MAXNI_SLOTS * ((64 * TCAM Slice Value) – 4)	NO	NA	NA
22	Number of Media streams per system in case link aggregation as edge port.	(64 * TCAM Slice value) - 4	NO	NA	NA
23	Logging Number of calls	200	YES	50	500

a. Subtracted value of 4 is due to the 15 UDF entries required for SIP method based trapping.

Note. When the Jitter, Packet Lost, and RTT crosses the configured threshold traps are raised. And when the R-factor and MOS goes below the configured threshold traps are raised.

Quick Steps for Configuring SIP Snooping

The following steps provide a quick tutorial on how to configure SIP Snooping. Each step describes a specific operation and provides the CLI command syntax for performing that operation.

1 Create a global SIP policy to classify incoming flows. Use the **policy condition** command to create a QOS condition. For example,

```
-> policy condition Voice sip audio
-> policy condition Video sip video
```

2 Create a policy action for the SIP condition using the **policy action** command. For example,

```
-> policy action DSCP46 dscp 46
-> policy action DSCP32 dscp 32
```

3 Configure the policy rule for the SIP policy to classify inbound and outbound media streams. Use the **policy rule** command. For example,

```
-> policy rule Voice_46 condition Voice action DSCP46
-> policy rule Video_32 condition Video action DSCP32
-> qos apply
```

Note. For more information on policy condition, policy action, and policy rule, see "**Configuring QOS**" chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

---

4 Enable SIP Snooping using the **sip-snooping admin-state** command. For example:

```
-> sip-snooping admin-state enable
```

This command will enable SIP snooping globally.

Note. When SIP snooping is enabled globally the SIP snooping is enabled on all ports and linkagg. The user can disable SIP snooping on specific port or linkagg (see Step 5).

---

5 Configure port/linkagg level SIP Snooping for the switch. Use the **sip-snooping admin-state** command with the **port** or **linkagg** parameter. For example,

```
-> sip-snooping port 1/1/5-6 admin-state enable
-> sip-snooping linkagg 1/1/10 admin-state enable
```

6 Configure the port/linkagg mode to force-edge for the port to which the SIP phone is connected. Use the **sip-snooping mode** command. For example,

```
-> sip-snooping port 1/1/5-6 mode force-edge
-> sip-snooping linkagg 1/1/10 mode force-edge
```

7 Configure the port/linkagg mode to force-non-edge for uplink port connecting to the call server. Use the **sip-snooping mode** command. For example,

```
-> sip-snooping port 1/5-6 mode force-non-edge
-> sip-snooping linkagg 1/1/10 mode force-non-edge
```

SIP Snooping Overview

The Session Initiation Protocol (SIP) is an IETF-defined signaling protocol widely used for controlling communication sessions such as voice and video calls over Internet Protocol (IP). The protocol can be used for creating, modifying and terminating media sessions. Sessions may consist of one or several media streams.

Other SIP applications include video conferencing, streaming multimedia distribution, instant messaging, presence information, file transfer and online games.

The SIP protocol is an Application Layer protocol designed to be independent of the underlying Transport Layer. SIP can run on the Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or Stream Control Transmission Protocol (SCTP). It is a text-based protocol, incorporating many elements of the Hypertext Transfer Protocol (HTTP) and the Simple Mail Transfer Protocol (SMTP).

The SIP Snooping feature is provided to address the key challenge of real time delivery and monitoring requirements for media streams from SIP devices. The feature allows automatic detection of SIP and its corresponding media streams.

The network is the most critical part of the enterprise infrastructure in delivering diverse applications. Ever increasing applications and their need for network resources keep demand on networks high.

- Critical applications like real-time voice, video, and mission critical data applications continue to grow.
- Bandwidth needs are growing at a faster pace than the network technologies that need to address them.
- Elastic traffic, such as TCP-based non-real time traffic, tends to use any additional bandwidth available.

It is essential to differentiate the various types of traffic, based on application, user, and context, and provide applicable service levels for each.

- Voice and video traffic should be prioritized over non-voice traffic.
- Mission critical data traffic should be provided a bandwidth guarantee for better performance.

The network should be able to monitor the quality of this traffic and inform the user if it is not within the required expectation. SIP Snooping addresses this issue for media streams managed by SIP.

The SIP snooping feature snoops voice quality metrics of media streams from their corresponding control packets and displays them to the user with knowledge of media reception quality in real time and helps to diagnose the problems on their quality. In addition, a trap is generated when the voice quality parameters crosses a user configured threshold.

Using SIP Snooping

A SIP network consists of the following network elements:

- Edge switches, aggregation switches, and core switches
- SIP user agents (e.g., SIP phones). SIP user agents are directly connected to edge switches

One SIP Server is connected to the Core switch within the campus infrastructure The server is responsible for all the SIP functions such as registrar, proxy, redirect, gateway.

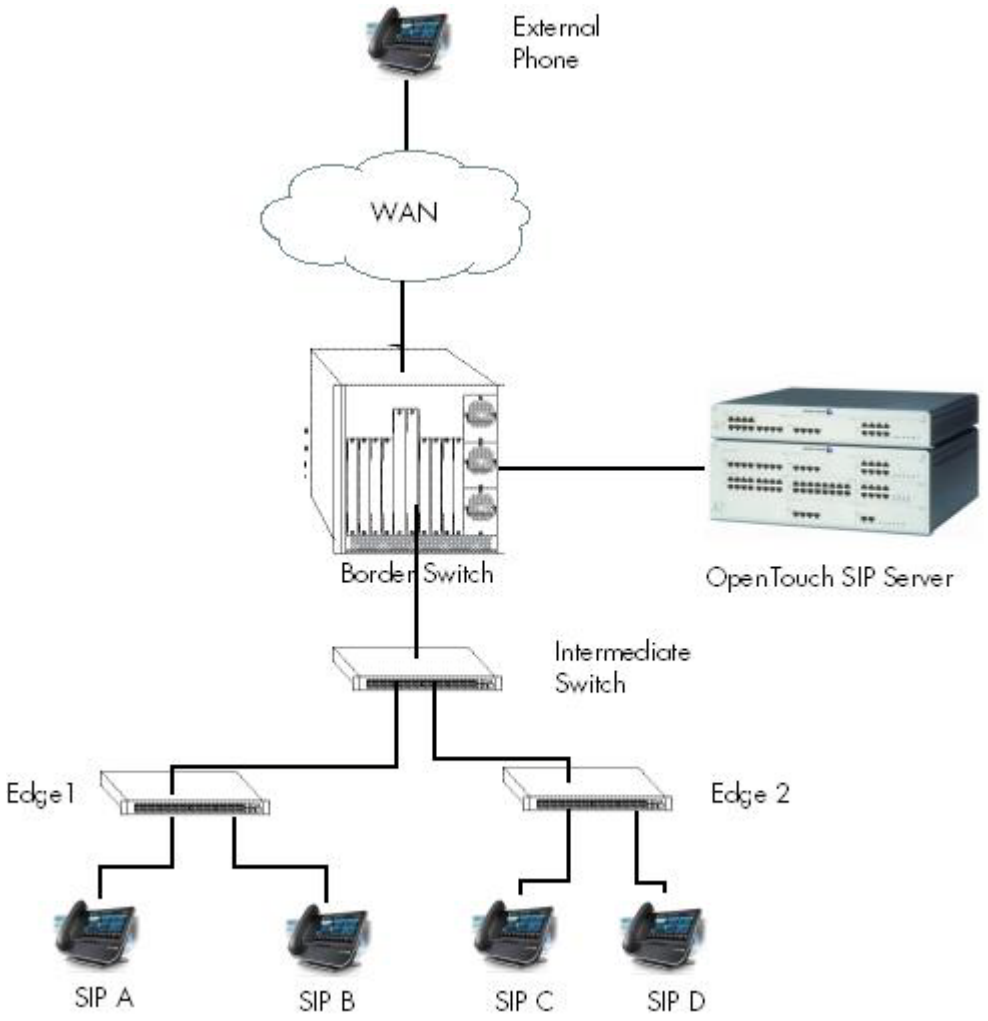


Figure 16-1 : Using SIP Snooping

In the above network, SIP-snooping is enabled on the edge switches.

- For an internal call, QOS treatment is enforced on both edge switches on which the SIP user agent endpoints are connected. On each edge switch, the QOS treatment is enforced for both ingress and egress media streams.
- For an external call, QOS treatment is only enforced on the edge switch on which the internal SIP user agent endpoint is connected. The media streams traversing the aggregation and core switches will not be subject to the SIP QOS treatment. On the edge switch, the QOS treatment is enforced for both ingress and egress media streams.

Interoperability

SIP Snooping can interact with the following equipment:

No	Equipment	Description
1	OpenTouch Business Edition 1.1 Server 500 Users (OTBE)	SIP based server from Alcatel-Lucent Enterprise
2	OXE IP Media Gateway MR3	Part of OTBE
3	PCX Enterprise RM3	Part of OTBE
4	Open Touch soft-phone - My Instant Communicator	Part of OTBE
5	8082 My IC Phone	OpenTouch SIP Phone
6	LifeSize Passport(Model: LFZ014)	SIP endpoint

SIP Snooping Configuration Guidelines

This section describes how to use OmniSwitch Command Line Interface (CLI) commands to configure SIP Snooping on a switch. Consider the following guidelines when configuring SIP Snooping entities:

Configuring Edge Port

SIP snooping requires that the uplink ports are configured as non-edge port. An edge port is a port on which the SIP user agent is connected. A non-edge port is the uplink port on which no SIP user agent is connected but requires SIP snooping. All AOS features available for an edge port are supported with SIP snooping.

By default, the edge and non-edge port modes are implicitly based on LLDP.

- A port that learns a LLDP remote agent advertising the “switch/router” capability is considered a non-edge port.
- A port that does not learn a LLDP remote agent advertising the “switch/router” capability is considered an edge port. A port can be forced to the edge or non-edge mode.

To configure the force-edge/force-non-edge, use the command as follows.

```
-> sip-snooping port 1/1/5-6 mode force-edge
-> sip-snooping port 1/1/10 mode force-non-edge
```

On the edge switch, it is recommended to disable auto phone with the **qos no phones** command. For example

```
-> qos no phones
```

Set all edge ports, including network edge ports to the un-trusted mode. Also ensure uplink port and all the traversing ports to other SIP endpoint are in trust mode.

Configuring Trusted SIP Server

By default, any SIP server is trusted. The SIP messages are trusted regardless of the origin (e.g., source IP address) or destination (e.g., destination IP address) of the SIP message.

The SIP snooping feature allows the configuration of trusted SIP servers. This restricts the SIP snooping functions to only a list of trusted server IP address.

Up to 8 trusted addresses can be configured as trusted SIP server. For configuring the trusted SIP server, use the command

```
-> sip-snooping trusted-server 192.168.0.1
```

All SIP based calls using other than configured trusted call server will not be subject to the configured SIP QOS treatment

Configuring SIP Snooping TCP Ports

The SIP snooping feature allows the configuration of TCP ports. This allows the SIP snooping functions to a list of TCP ports, SIP packets sent/received on the TCP ports will be snooped. A maximum of 8 TCP ports can be configured on a switch.

To configure the Server listening TCP ports, use the **sip-snooping TCP port** as follows

```
-> sip-snooping tcp-port 5678 5051
```

The SIP Snooping TCP port configuration is used to snoop the SIP packets, when the SIP endpoints switches from UDP to TCP to send SIP packets of more than 1300 bytes in size.

Configuring SIP Snooping UDP Ports

The SIP snooping feature allows the configuration of UDP ports. This allows the SIP snooping functions to a list of UDP ports, SIP packets sent/received on the UDP ports will be snooped. A maximum of 8 UDP ports can be configured on a switch.

To configure the Server listening UDP port, use the **sip-snooping UDP port** as follows

```
-> sip-snooping udp-port 5260 5060
```

This allows the SIP snooping functionality for the configured UDP ports only.

Configuring the SIP Control DSCP

To configure SIP control DSCP marking, use the **sip-snooping sip-control** command

```
-> sip-snooping sip-control dscp 40
```

This marks the SIP messages with the configured SIP control DSCP.

Configuring SOS Calls

The SIP snooping features allow the detection of emergency calls based on the “to” URI in the INVITE message. Configuration allows up to 4 SOS call strings to be configured. The string must be the exact URI to be matched in the ‘to” URI.

When a call is deemed to be a SOS call, a default DSCP of 46 (EF) is assigned for both RTP and RTCP flows of that call. SOS-Call DSCP can be configured to any value.A non-configurable rate limiter of 128kbps is imposed for SOS-Call.

```
-> sip-snooping sos-call number “911” “2233”
```

By default, no SOS number is configured for SIP Snooping

Configuring SOS Call DSCP

To configure the SOS-Call RTP/RTCP DSCP Marking, use the **sip-snooping sos-call** command.

```
-> sip-snooping sos-call dscp 56
```

This marks the SOS-Call RTP/RTCP packets with the configured SOS call dscp.

Configuring RTCP Thresholds

When RTCP monitoring is enabled, the SIP snooping feature also inspects the RTCP packet that carries performance metric for the RTP flow.

Depending on the RTCP capabilities of the SIP user agent endpoints, the following metrics can be determined by software:

- Packet loss
- Round Trip Time
- R Factor Only supported with RTCP-XR
- MOS factor – Only supported with RTCP-XR

The SIP snooping feature offers configurable thresholds for each performance metric and each media types.

```
-> sip-snooping threshold audio jitter 30
-> sip-snooping threshold video jitter 50
-> sip-snooping threshold audio packet-lost 40
-> sip-snooping threshold video packet-lost 30
-> sip-snooping threshold audio round-trip-delay 180
-> sip-snooping threshold video round-trip-delay 180
-> sip-snooping threshold audio R-factor 30
-> sip-snooping threshold video R-factor 30
-> sip-snooping threshold audio MOS 1
-> sip-snooping threshold video MOS 2
```

Configuring the Logging Threshold for the Number of Calls

To configure the threshold for the number of call records to be logged on to the flash file, use the **sip-snooping logging-threshold num-of-calls** command as follows

```
-> sip-snooping logging-threshold num-of-calls 300
```

Configuring Policy Rules for SIP Snooping

Unlike regular policy rule, a SIP policy rule is not programmed in the hardware when it is configured. The ACL is only programmed when the SIP snooping module detects a new RTP flow and parses the SIP policy rules to determine the QOS policy actions required for this RTP flow.

Policy Condition

All other policy conditions are still supported for the SIP policy rules. This allows specific QOS treatments (policy actions) for media streams based on the origin of the call. For instance, a SIP policy condition can be combined with:

- Source port
- Source IP address/subnet

To configure the policy condition, use the commands as follows.

```
-> policy condition <condition_name> sip {audio | video | other}
-> policy condition <condition_name> sip {audio | video | other}source port 1/2
```

Other source conditions are also supported but are not foreseen to provide real benefits.

The policy condition is not used as such in the hardware filtering entry, but is used by the SIP snooping module to determine the policy rule that the new RTP flow is matching. Also, SIP policy rules are supported in ingress and UNP policy lists.

Policy Action

All other policy actions are still supported for SIP policy rules. For instance:

- DSCP—marks the DSCP value for the RTP/RTCP packets and maps the internal priority to this DSCP
- Priority—forces the internal priority of the RTP/RTCP packets.
- Rate Limiter

To configure the policy action, use the commands as follows.

```
-> policy action <action_name> dscp 32 rtcp-monitoring {enable | disable}
-> policy action <action_name> dscp 46 rtcp-monitoring enable rtcp-dscp <num>
-> policy action <action_name> rtcp-monitoring disable trust-dscp
```

Policy Rule

A SIP policy rule is a rule that has the keyword SIP (audio/video/other) in the policy condition and a corresponding policy action.

The SIP policy rule is configured as follows.

```
-> policy condition Voice_srcip_SIP1 source ip 10.10.0.0 mask 255.255.0.0 sip
audio
-> policy condition Video_srcip_SIP1 source ip 10.10.0.0 mask 255.255.0.0 sip
video
-> policy action DSCP56 dscp 56
-> policy action DSCP32 dscp 32
-> policy rule Voice_srcip_SIP1_rule condition Voice_srcip_SIP1 action DSCP56
-> policy rule Video_srcip_SIP1_rule condition Video_srcip_SIP1 action DSCP32
-> qos apply
```

Note that a SIP policy rule does not apply for the SIP signaling messages. A SIP policy rule can however apply for a SOS call since a SOS call is a audio media. However, the DSCP marking and rate limiter for an SOS call cannot be overwritten by a SIP policy rule.

Unsupported Topologies

The SIP snooping functions and the QOS actions require that the network paths used by the SIP signaling messages and the RTP/RTCP flows are the same and are “symmetric”. For this reason, the following topologies are not supported:

- ECMP Routing
- VRRP

In such topologies, it would be possible that one switch/router sees the SIP signaling messages and creates the dialog with the appropriate QOS actions (i.e. ACLs), but the RTP/RTCP traffic will not be seen by this switch/router. It would also be possible that the SIP signaling messages and/or RTP/RTCP packets are load balanced between 2 switch/routers.

SIP Snooping Use Case

In this section, advanced SIP configuration use cases are illustrated. Instead of having all voice audio/ video media streams treated the same way, more granular SIP policies can be configured.

Expectations

- Voice media streams from SIP1 to SIP2 will be marked with DSCP 56
- Video media streams from Video SIP1 to Video SIP2 will be marked with DSCP 32
- Voice media streams from SIP2 to SIP1 will be marked with DSCP 46
- Voice media streams from Video SIP2 to Video SIP1 will be marked with DSCP 48

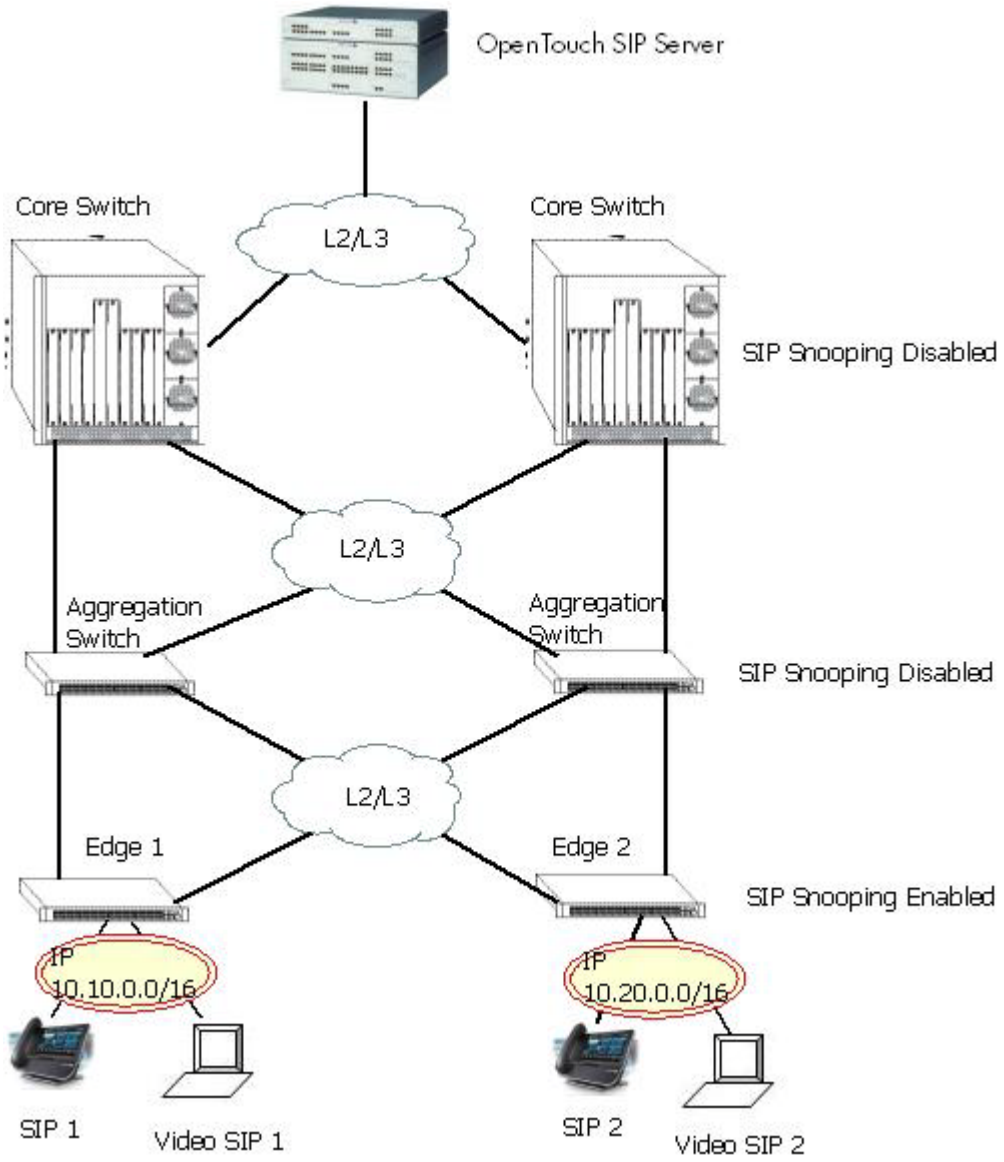


Figure 16-2 : SIP Snooping Use Case

SIP Condition

In this example, specific QoS treatments are configured based on the source IP subnet.

- Voice source IP subnet 10.10.0.0 = DSCP 56
- Video source IP subnet 10.10.0.0=DSCP 32
- Voice source IP subnet 10.20.0.0 = DSCP 46
- Video source IP subnet 10.20.0.0=DSCP 48

The SIP conditions are configured as follows:

```
-> policy condition Voice_srcip_SIP1 source ip 10.10.0.0 mask 255.255.0.0 sip
audio
-> policy condition Video_srcip_SIP1 source ip 10.10.0.0 mask 255.255.0.0 sip
video
-> policy condition Voice_srcip_SIP2 source ip 10.20.0.0 mask 255.255.0.0 sip
audio
-> policy condition Video_srcip_SIP2 source ip 10.20.0.0 mask 255.255.0.0 sip
video
-> policy action DSCP56 dscp 56
-> policy action DSCP32 dscp 32
-> policy action DSCP46 dscp 46
-> policy action DSCP48 dscp 48
-> policy rule Voice_srcip_SIP1_rule condition Voice_srcip_SIP1 action DSCP56
-> policy rule Video_srcip_SIP1_rule condition Video_srcip_SIP1 action DSCP32
-> policy rule Voice_srcip_SIP2_rule condition Voice_srcip_SIP2 action DSCP46
-> policy rule Video_srcip_SIP2_rule condition Video_srcip_SIP2 action DSCP48
-> qos apply
```

The active call records can be viewed by using the following command:

```
-> show sip-snooping call-records active-calls full
Legend: start date time duration media-type end-reason
         call-id / from-tag / to-tag
         IP address port DSCP (forward/reverse)
         policy-rule (F/R)

   statistics min / max / avg %samples exceeding threshold (F/R)
-------------------------------------------------------------------------------
2013-11-21 18:39:17(PST) 0d 16h 13m 41s Audio    -
6dddf3236f2d564c / d1fc26f8da / 0061D0A0-7C50-1200-83AF-F1A3FE87AA77-1439499
IP/DSCP         5.5.5.2 6000 NA/NA      7.7.7.2 6000 NA/NA
Policy-Rule     r6      r1
Pkt-Count       2920807 2920807
Pkt-Loss        0       0          0.00            0%   0       0    0.00     0%
Jitter          1       198714     17.34           0%   1       49   0.32     0%
Delay           9       29         16.44           0%   9       29   16.44    0%
R-factor        35      96         35.42          99%   30      96   32.00    99%
MOS             1.00    4.00       1.80           99%   1.00    4.00 1.60     99%

-------------------------------------------------------------------------------

Number of Call Records: 1
```

```
-> show sip-snooping call-records ended-calls full
Legend: start date time duration media-type end-reason
         call-id / from-tag / to-tag
         IP address port DSCP (forward/reverse)
         policy-rule (F/R)
         Pkt count (F/R)

            statistics min / max / avg %samples exceeding threshold (F/R)
-------------------------------------------------------------------------------
2002-04-06 01:06:10 UTC 0d 0h 4m 15s    Audio    -
0010CFC0-4A05-10DA-B960-F1A3FE87AA77-23025@ot380.aos.com / 0010CFE8-4A05-10DA-B960-
F1A3FE87AA77-258649 / 1668946822
IP/DSCP         10.20.0.2 6000 56/56    10.10.0.2 6000 46/46
Policy-Rule     Voice_srcip_SIP1_rule   Voice_srcip_SIP2_rule
Pkt-Count       12272   61385
Pkt-Loss        0       0         0.00           0% 0       0         0.00
0%
Jitter          0       0         0.00           0% 0       0         0.00
0%
Delay           0       0         0.00           0% 0       0         0.00
0%
R-factor        0       0         0.00           0% 96      96        96.00
0%
MOS              0       0         0.00           0% 44      44        44.00

-------------------------------------------------------------------------------
Number of Call Records: 1
```

Similar to the above example, more conditions can be combined in a single SIP rule.

Advanced RTCP Control

For each RTP flow, RTCP monitoring can be enabled or disabled. When enabled, the DSCP marking can also be controlled. Also Trap will be generated if RTCP parameters exceed the Threshold values configured in SIP configuration.

In this example, specific QOS treatments are configured based on the Source IP subnet.

- Voice source IP subnet 10.10.0.0 = DSCP 56— RTCP packets for these RTP flows are trapped to CPU and assigned with DSCP 56.

  ```
  -> policy action DSCP56 dscp 56
  ```

- Video source IP subnet 10.10.0.0= RTCP—packets for these RTP flows are trapped to CPU and have their DSCP unchanged.

  ```
  -> policy action DSCP32 rtcp-monitoring enable trust-DSCP
  ```

- Voice source IP subnet 10.20.0.0 = DSCP 46 + No RTCP monitoring—RTCP packets for these RTP flows are not trapped to CPU and assigned with DSCP 46.

  ```
  -> policy action DSCP46 dscp 46 rtcp-monitoring disable
  ```

- Video source IP subnet 10.20.0.0 = DSCP 48 + RTCP monitoring and explicit DSCP 46—RTCP packets for these RTP flows are trapped to CPU and assigned with DSCP 46.

  ```
  -> policy action DSCP48 dscp 48 rtcp-monitoring enable rtcp-dscp 46
  ```

SIP Snooping Limitations

- Media types other than audio and video as application, image media types etc. are not supported.
- Solution only supports SIP, no support of NOE (New Office Environment).
- SIP Registrar, outbound proxy, proxy, redirect functions should be provided by the same server called the SIP Server.
- All initial SIP messages between User Agents must go through the SIP Server. Direct SIP session establishment between end users will be not supported.
- Outbound proxy configured on phone and trusted call server configured on switch must be the same.
- Only SIP over UDP and SIP over UDP/TCP is supported. Solution does not support SIP over SCTP or MPLS and SIP over TLS.
- Encrypted RTCP or SDP is not supported.
- Only SIP over IPv4 is supported, no support for IPV6.
- Multicast Media Sessions by SIP is not supported
- Only RTP or RTP profile AVP is supported to carry media. SAVP, AVPF, SAVPF are not supported.
- Only IP address is supported. DNS resolution and FQDN name are not supported in SDP
- Only audio and video application in "m" line of SDP is supported.
- No network performance reporting other than RTCP reports.
- RTCP port assignment is taken as one higher than corresponding RTP. Other methods for RTCP port assignment is not supported
- Media quality metrics displayed to the user only convey the presence of problem in voice and video transmission quality. Exact location and device responsible for it will not be known and it is expected that the user will find it by other means and take corrective action.
- QOS SIP policy modifications should be applied for the new calls only and not for existing ones.
- DSCP marking will be done for only [(64 * TCAM Slice Value)-4] SIP audio calls, if a call is through linkagg on a stack.
- No VRF awareness. Similarly, NAT transversal (ICE, TURN, STUN solution) is not supported.
- Emergency call identification is based on user configured string. Usage of priority or resource-priority header is not considered.
- SIP IP address and RTP IP address of end point at edge port must be same, otherwise TCAM entries will not be created.
- Media that flows before TCAM entries are installed does not get configured QOS treatment.

Verifying the SIP Snooping Configuration

To display information about Sip Snooping on the switch, use the show commands listed below:

show sip-snooping config	Shows the SIP snooping configuration.
show sip-snooping ports	Displays the SIP snooping port level data.
show sip-snooping call-records	Displays the SIP-snooping active/ended call records.
show sip-snooping statistics	Displays the SIP snooping statistics.

17 Configuring IP

Internet Protocol (IP) is primarily a network-layer (Layer 3) protocol that contains addressing and control information that enables packets to be forwarded. Along with Transmission Control Protocol (TCP), IP represents the heart of the Internet protocols. IP has two primary responsibilities, providing connectionless, best-effort delivery of datagrams through an internetwork; and providing fragmentation and reassembly of datagrams to support data links with different Maximum Transmission Unit (MTU) sizes.

Note. IP routing (Layer 3) can be accomplished using static routes or by using one of the IP routing protocols, Routing Information Protocol (RIP) and Open Shortest Path First (OSPF). For more information on these protocols see Chapter 21, "Configuring RIP," in this manual; or "Configuring OSPF" in the *OmniSwitch AOS Release 8 Advanced Routing Configuration Guide*.

Two versions of Internet Protocol are supported - IPv4 and IPv6. For more information about using IPv6, see Chapter 19, "Configuring IPv6."

In This Chapter

This chapter describes IP and how to configure it through the Command Line Interface (CLI). It includes instructions for enabling IP forwarding, configuring IP route maps, as well as basic IP configuration commands (for example, **ip default-ttl**). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. This chapter provides an overview of IP and includes information about the following procedures:

- IP Forwarding
  - Configuring an IP Interface (see page 17-7)
  - Configuring an IP Managed Interface (see page 17-10)
  - Creating a Static Route or Recursive Static Route (see page 17-11)
  - Creating a Default Route (see page 17-12)
  - Configuring a Blackhole Route (see page 17-12)
  - Configuring an IP Routed Port (see page 17-13)
  - Configuring Address Resolution Protocol (ARP) (see page 17-13)
- IP Configuration
  - Configuring the Router Primary Address (see page 17-19)
  - Configuring the Router ID (see page 17-19)
  - Configuring the Time-to-Live (TTL) Value (see page 17-20)
  - Configuring Route Map Redistribution (see page 17-20)
  - IP-Directed Broadcasts (see page 17-26)
  - Protecting the Switch from Denial of Service (DoS) attacks (see page 17-27)

- Managing IP
  - Internet Control Message Protocol (ICMP) (see page 17-33)
  - Using the Ping Command (see page 17-35)
  - Tracing an IP Route (see page 17-36)
  - Transmission Control Protocol (TCP) (see page 17-36)
  - Displaying User Datagram Protocol (UDP) Information (see page 17-37)
  - Service Assurance Agent (SAA) (see page 17-37)
- Tunneling
  - Generic Routing Encapsulation (page 17-37)
  - IP Encapsulation within IP (page 17-37)
  - Tunneling operation (page 17-38)
  - Configuring a Tunnel Interface (page 17-39)
- VRF Route Leak
  - Quick Steps for Configuring VRF Route Leak (page 17-41)
  - Configuring VRF Route Leak (page 17-42)
  - Verifying VRF Route Leak Configuration (page 17-43)

IP Defaults

The following table lists the defaults for IP configuration through the **ip** command.

Description	Command	Default
IP-Directed Broadcasts	**ip directed-broadcast**	disable
Time-to-Live Value	**ip default-ttl**	64 (hops)
IP interfaces	**ip interface**	VLAN 1 interface.
ARP filters	**arp filter**	0

Quick Steps for Configuring IP Forwarding

Using only IP, which is always enabled on the OmniSwitch, devices connected to ports on the same VLAN are able to communicate at Layer 2. The initial configuration for all OmniSwitch platforms consists of a default VLAN 1. All switch ports are initially assigned to this VLAN. If additional VLANs are not configured on the switch, the entire switch is treated as one large broadcast domain, and all ports receive all traffic from all other ports.

Note. The operational status of a VLAN remains inactive until at least one active switch port is assigned to the VLAN. If the ports are connected to an active network device, they are considered active. Non-active port assignments are allowed, but do not change the operational state of the VLAN.

To forward packets to a different VLAN on a switch, create an IP interface on each VLAN. The following steps provide a quick tutorial of how to enable IP forwarding between VLANs “from scratch”. If active VLANs have already been created on the switch, you only need to create IP interfaces on each VLAN (Steps 5 and 6).

1 Create VLAN 10 with a description (for example, VLAN 10) using the **vlan** command. For example:

```
-> vlan 10 name "VLAN 10"
```

2 Create VLAN 20 with a description (for example, VLAN 20) using the **vlan** command. For example:

```
-> vlan 20 name "VLAN 20"
```

3 Assign an active port to VLAN 10 using the **vlan members untagged** command. For example, the following command assigns port 1 on slot 1 to VLAN 10:

```
-> vlan 10 members port 1/1 untagged
```

4 Assign an active port to VLAN 20 using the **vlan members** command. For example, the following command assigns port 2 on slot 1 to VLAN 20:

```
-> vlan 20 members port 1/2 untagged
```

5 Create an IP interface on VLAN 10 using the **ip interface** command. For example:

```
-> ip interface vlan-10 address 171.10.1.1 vlan 10
```

6 Create an IP interface on VLAN 20 using the **ip interface** command. For example:

```
-> ip interface vlan-20 address 171.11.1.1 vlan 20
```

Note. See Chapter 4, "Configuring VLANs," for more information about how to create VLANs.

IP Overview

IP is a network-layer (Layer 3) protocol that contains addressing and control information that enables packets to be forwarded on a network. IP is the primary network-layer protocol in the Internet protocol suite. Along with TCP, IP represents the heart of the Internet protocols.

IP Protocols

IP is associated with Layer 3 and Layer 4 protocols. These protocols are built into the base code loaded on the switch. A brief overview of the supported IP protocols is described in the following sections.

Transport Protocols

IP is both connectionless (it forwards each datagram separately) and unreliable (it does not guarantee delivery of datagrams). This means that a datagram can be damaged in transit, thrown away by a busy switch, or never make it to its destination. The resolution of these transit problems is to use a Layer 4 transport protocol, such as:

- TCP—A major data transport mechanism that provides reliable, connection-oriented, full-duplex data streams. While the role of TCP is to add reliability to IP, TCP relies upon IP to do the actual delivering of datagrams.
- UDP—A secondary transport-layer protocol that uses IP for delivery. UDP is not connection-oriented and does not provide reliable end-to-end delivery of datagrams. Few applications can safely use UDP to send datagrams that do not require the extra overhead added by TCP. For more information on UDP, see Chapter 23, "Configuring DHCP Relay."

Application-Layer Protocols

Application-layer protocols are used for switch configuration and management:

- Bootstrap Protocol (BOOTP)/Dynamic Host Configuration Protocol (DHCP)—can be used by an end station to obtain an IP address. The switch provides a DHCP Relay that allows BOOTP requests/replies to cross different networks.
- Simple Network Management Protocol (SNMP)—Allows communication between SNMP managers and SNMP agents on an IP network. Network administrators use SNMP to monitor network performance and manage network resources. For more information, see the "Using SNMP" chapter in the *OmniSwitch AOS Release 8 Switch Management Guide*.
- Telnet—Used for remote connections to a device. You can telnet to a switch and configure the switch and the network by using the CLI.
- SSH—Used for remote connections to a device. You can SSH to a switch and configure the switch and the network by using the CLI.

- File Transfer Protocol (FTP)—Enables the transfer of files between hosts. This protocol is used to load new images onto the switch.

Additional IP Protocols

Many additional IP-related protocols can be used with IP forwarding. These protocols are included as part of the base code.

- **Address Resolution Protocol (ARP)**—Used to match the IP address of a device with its physical (MAC) address. For more information, see "Configuring Address Resolution Protocol (ARP)" on page 17-13.
- Virtual Router Redundancy Protocol (VRRP)—Used to back up routers. For more information, see Chapter 25, "Configuring VRRP."
- Internet Control Message Protocol (ICMP)—Specifies the generation of error messages, test packets, and informational messages related to IP. ICMP supports the **ping** command used to determine if hosts are online. For more information, see "Internet Control Message Protocol (ICMP)" on page 17-33.
- Multicast Services—Includes IP multicast switching (IPMS). For more information, see Chapter 27, "Configuring IP Multicast Switching."

IP Forwarding

Network device traffic is bridged (switched) at the Layer 2 level between ports that are assigned to the same VLAN. However, if a device needs to communicate with another device that belongs to a different VLAN, then Layer 3 routing is necessary to transmit traffic between the VLANs. Bridging decides to forward the packets based on the destination MAC address of the packet. Routing decides on where to forward packets based on the IP network address of the packet (for example, IP - 21.0.0.10).

The OmniSwitch platforms support routing of IP traffic. A VLAN is available for routing when at least one- router interface is defined for that VLAN and at least one active port is associated with the VLAN. If a VLAN does not have a router interface, the ports associated with that VLAN are in essence firewalled from other VLANs.

IP multi-netting is also supported. A network is said to be multi-netted when multiple IP subnets are brought together within a single broadcast domain. Each interface is configured with a different subnet. As a result, traffic from each configured subnet can coexist on the same VLAN.

In the illustration below, an IP router interface has been configured on each VLAN. Therefore, workstations connected to ports on VLAN 1 on Switch 1 can communicate with VLAN 2; and workstations connected to ports on VLAN 3 on Switch 2 can communicate with VLAN 2. Also, ports from both switches have been assigned to VLAN 2, and a physical connection has been made between the switches. Therefore, workstations connected to VLAN 1 on Switch 1 can communicate with workstations connected to VLAN 3 on Switch 2.

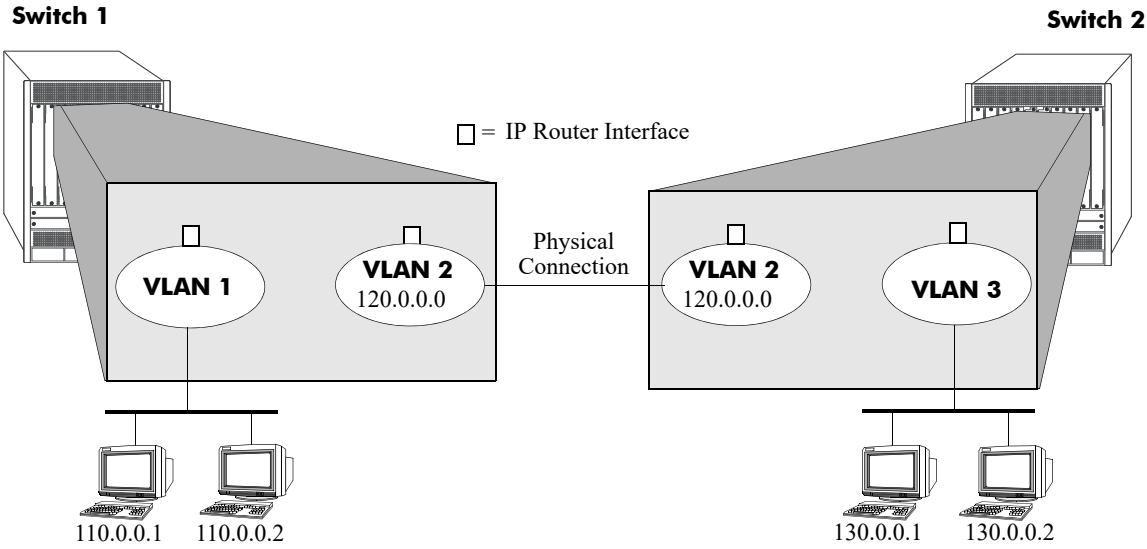


Figure 17-1 : IP Forwarding

Configuring an IP Interface

IP is enabled by default. Using IP, devices connected to ports on the same VLAN are able to communicate. However, to forward packets to a different VLAN, create at least one IP interface on each VLAN.

Use the **ip interface** command to define IP interfaces for an existing VLAN. The following parameter values are configured with this command:

- A unique interface name (text string up to 16 characters) is used to identify the IP interface. Specifying this parameter is required to create or modify an IP interface.
- The VLAN ID of an existing VLAN.
- An IP address to assign to the interface (for example, 193.204.173.21). Router interface IP addresses must be unique. You cannot have two-router interfaces with the same IP address.
- A subnet mask (defaults to the IP address class). It is possible to specify the mask in dotted decimal notation (for example, 255.255.0.0) or with a slash (/) after the IP address followed by the number of bits to specify the mask length (for example, 193.204.173.21/24).
- The forwarding status for the interface (defaults to forwarding). A forwarding router interface sends IP frames to other subnets. A router interface that is not forwarding can receive frames from other hosts on the same subnet.
- An Ethernet-II or SNAP encapsulation for the interface (defaults to Ethernet-II). The encapsulation determines the framing type the interface uses when generating frames that are forwarded out of VLAN ports. Select an encapsulation that matches the encapsulation of the majority of VLAN traffic.
- The Local Proxy ARP status for the VLAN. If enabled, traffic within the VLAN is routed instead of bridged. ARP requests return the MAC address of the IP router interface defined for the VLAN. For more information about Local Proxy ARP, see "Local Proxy ARP" on page 17-15.
- The primary interface status. Designates the specified IP interface as the primary interface for the VLAN. By default, the first interface bound to a VLAN becomes the primary interface for that VLAN.

The following **ip interface** command example creates an IP interface named Marketing with an IP network address of 21.0.0.1 and binds the interface to VLAN 455:

```
-> ip interface Marketing address 21.0.0.1 vlan 455
```

The **name** parameter is the only parameter required with this command. Specifying additional parameters is only necessary to configure a value other than the default value for that parameter. For example, all of the following commands create an IP router interface for VLAN 955 with a class A subnet mask, an enabled forwarding status, Ethernet-II encapsulation, and a disabled Local Proxy ARP and primary interface status:

```
-> ip interface Accounting address 71.0.0.1 mask 255.0.0.0 vlan 955 forward e2
no local-proxy-arp no primary
-> ip interface Accounting address 71.0.0.1/8 vlan 955
-> ip interface Accounting address 71.0.0.1 vlan 955
```

Modifying an IP Router Interface

The **ip interface** command is also used to modify existing IP interface parameter values. It is not necessary to remove the IP interface and then create it again with the new values. The changes specified overwrite existing parameter values. For example, the following command changes the subnet mask to **255.255.255.0**, the forwarding status to **no forwarding** and the encapsulation to **snap** by overwriting existing parameter values defined for the interface. The interface name, **Accounting**, is specified as part of the command syntax to identify which interface to change.

```
-> ip interface Accounting mask 255.255.255.0 no forward snap
```

When changing the IP address for the interface, the subnet mask reverts to the default mask value if it was previously set to a non-default value and it is not specified when changing the IP address. For example, the following command changes the IP address for the Accounting interface:

```
-> ip interface Accounting address 40.0.0.1
```

The subnet mask for the Accounting interface was previously set to 255.255.255.0. The above example resets the mask to the default value of 255.0.0.0 because 40.0.0.1 is a Class A address and no other mask was specified with the command. This only occurs when the IP address is modified; all other parameter values remain unchanged unless otherwise specified.

To avoid the problem in the above example, enter the non-default mask value whenever the IP address is changed for the interface. For example:

```
-> ip interface Accounting address 40.0.0.1 mask 255.255.255.0
-> ip interface Accounting address 40.0.0.1/8
```

Use the **show ip interface** command to verify IP router interface changes. For more information about these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide.*

Removing an IP Router Interface

To remove an IP router interface, use the **no** form of the **ip interface** command. It is only necessary to specify the name of the IP interface, as shown in the following example:

```
-> no ip interface Marketing
```

To view a list of IP interfaces configured on the switch, use the **show ip interface** command. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide.*

Configuring a Loopback0 Interface

Loopback0 is the name assigned to an IP interface to identify a consistent address for network management purposes. The Loopback0 interface is not bound to any VLAN, so it always remains operationally active. If there are no active ports in the VLAN, all IP interface associated with that VLAN are not active. In addition, the Loopback0 interface provides a unique IP address for the switch that is easily identifiable to network management applications.

This type of interface is created in the same manner as all other IP interfaces, using the **ip interface** command. To identify a Loopback0 interface, enter **Loopback0** for the interface name. For example, the following command creates the Loopback0 interface with an IP address of 10.11.4.1:

```
-> ip interface Loopback0 address 10.11.4.1
```

Note the following when configuring the Loopback0 interface:

- The interface name, “Loopback0”, is case sensitive.
- The Loopback0 interface is always active and available.
- Only one Loopback0 interface per switch is allowed.
- Loopback0 address cannot be modified once it is configured.
- Creating this interface does *not* deduct from the total number of IP interfaces allowed per VLAN or switch.
- To change the address, remove the interface using the **no ip interface Loopback0** command and add it again with the new address.

Loopback0 Address Advertisement

The Loopback0 IP interface address is automatically advertised by the IGP protocols RIP and OSPF when the interface is created. There is no additional configuration necessary to trigger advertisement with these protocols.

Note the following regarding Loopback0 advertisement:

- RIP advertises the host route to the Loopback0 IP interface as a redistributed (directhost) route.
- OSPF advertises the host route to the Loopback0 IP interface in its Router-LSAs (as a Stub link) as an internal route into all its configured areas.

Configuring a BGP Peer Session with Loopback0

It is possible to create BGP peers using the Loopback0 IP interface address of the peering router and binding the source (that is, outgoing IP interface for the TCP connection) to its own configured Loopback0 interface. The Loopback0 IP interface address can be used for both Internal and External BGP peer sessions. For EBGP sessions, if the external peer router is multiple hops away, the **ebgp-multihop** parameter can be used.

The following example command configures a BGP peering session using a Loopback0 IP interface address:

```
-> ip bgp neighbor 2.2.2.2 update-source Loopback0
```

See the *OmniSwitch AOS Release 8 Advanced Routing Configuration Guide* for more information.

Configuring an IP Managed Interface

By default, most applications that run on IP use the egress IP interface address as the source IP, while using a socket to communicate with a peer/server. However, it may be desirable to have some applications use a specific source IP for the packets that are sent out using the socket.

The **ip service source-ip** command provides the ability to configure a permanent source IP interface to send packets. The source IP interface can be the Loopback0 address or user defined IP interface. For example,

The following commands create a Loopback0 interface and configures that interface as a source IP interface for the FTP application:

```
-> ip interface Loopback0 address 10.10.1.1
-> ip service source-ip loopback0 ftp
```

The following command configures user-defined source IP interface for the FTP application:

```
-> ip service source-ip ipVlan100 ftp
```

Notes.

- Use the "**all**" option in the command to configure a common source IP address for the applications. If for a particular application, specific source IP address is configured and the "all" option is also set, the configured source IP address for the application is used as the outgoing interface.
- If a source IP interface is not defined for an application, the application uses the outgoing interface address as the source IP.

A source IP address is configurable for the following applications within the VRF context:

Application	Default Source Interface	VRF Support
ASA Authentication Server		
LDAP Server	Outgoing interface	Supported with any VRF (Configuration available only in the default VRF)
TACACS+	Outgoing interface	Supported with any VRF (Configuration available only in the default VRF)
AAA Authentication Server		
RADIUS	Outgoing interface	Supported with any VRF (Configuration available only in the default VRF)
Switch Management Applications		
SNMP (includes traps)	Outgoing interface	Supported with any VRF (Configuration available only in the default VRF)
SFLOW	Outgoing interface	Supported with only default VRF
SWLOG	Outgoing interface	Supported with any VRF (Configuration available only in the default VRF)

Application	Default Source Interface	VRF Support
DNS	Outgoing interface	Servers can only be set in the default VRF
Switch Access and Utilities (ping and traceroute command can specify a source address as an optional parameter)		
Telenet	Outgoing interface	Supported with any VRF
FTP	Outgoing interface	Supported with any VRF
SSH (includes scp, sftp)	Outgoing interface	Supported with any VRF
TFTP	Outgoing interface	Supported with any VRF

Creating a Static Route or Recursive Static Route

Static routes are user-defined and carry a higher priority than routes created by dynamic routing protocols. That is, if two routes have the same metric value, the static route has the higher priority. Static routes allow you to define, or customize, an explicit path to an IP network segment, which is then added to the IP Forwarding table. Static routes can be created between VLANs to enable devices on these VLANs to communicate.

Use the **ip static-route** command to create a static route. Specify the destination IP address of the route as well as the IP address of the first hop (gateway/interface) used to reach the destination. For example, to create a static route to IP address 171.11.0.0 through gateway 171.11.2.1 with a tag of 3, you would enter:

```
-> ip static-route 171.11.1.0/24 gateway 171.11.2.1 tag 3
```

For example, to create a proxy ARP static route to IP address 171.11.0.0 through interface Int1you would enter:

```
-> ip static-route 171.11.1.0/24 interface Int1
```

If you want to use the natural subnet mask, the subnet mask is not required. By default, the switch imposes a natural mask on the IP address. In the above example, the Class B mask of 255.255.0.0 is implied. If you do not want to use the natural mask, enter a subnet mask. For example, to create a static route to IP address 10.255.11.0, enter the Class C mask of 255.255.255.0:

```
-> ip static-route 10.255.11.0 mask 255.255.255.0 gateway 171.11.2.1
```

Specifying the length of the mask in bits is also supported. For example, the above static route is also configurable using the following command:

```
-> ip static-route 10.255.11.0/24 gateway 171.11.2.1
```

When you create a static route, the default metric value of 1 is used. However, you can change the priority of the route by increasing its metric value. The lower the metric value, the higher the priority. This metric is added to the metric cost of the route. The metric range is 1 to 15. For example:

```
-> ip static-route 10.255.11.0/24 gateway 171.11.2.1 metric 5
```

Static routes do not age out of the IP Forwarding table; delete them from the table. Use the **no ip static route** command to delete a static route. Specify the destination IP address of the route as well as the IP address of the first hop (gateway). For example, to delete a static route to IP address 171.11.0.0 through gateway 171.11.2.1, you would enter:

```
-> no ip static-route 171.11.1.0/24 gateway 171.11.2.1
```

The IP Forwarding table includes routes learned through one of the routing protocols (RIP, OSPF, BGP, ISIS) as well as any static routes that are configured. Use the **show ip routes** command to display the IP Forwarding table.

Creating a Recursive Static Route

Recursive static routes are similar to the static routes described above. However, with a recursive static route the route to reach the gateway is learned through a dynamic routing protocol such as RIP or OSPF. If a better route to the gateway is learned, the path to a recursive route can be changed dynamically. This feature can be used to configure a uniformed static route for all routers on a network, but the path to reach the gateway can differ for each router. To create a recursive static route use the **follows** parameter:

```
-> ip static-route 171.11.1.0/24 follows 192.168.10.1
```

A route to the **192.168.10.1** address must be learned by a dynamic routing protocol for the recursive static route to be active.

Creating a Default Route

A default route can be configured for packets destined for networks that are unknown to the switch. Use the **ip static-route** command to create a default route. Specify a default route of 0.0.0.0 with a subnet mask of 0.0.0.0 and the IP address of the next hop (gateway). For example, to create a default route through gateway 171.11.2.1 you would enter:

```
-> ip static-route 0.0.0.0 mask 0.0.0.0 gateway 171.11.2.1
```

Specifying the length of the mask in bits is also supported. For example, the above default route is also configurable using the following command:

```
-> ip static-route 0.0.0.0/0 gateway 171.11.2.1
```

Note. You cannot create a default route by using the EMP port as a gateway.

Configuring a Blackhole Route

A blackhole route is used to forward unwanted traffic to a black-hole. Static routes may be created for undesirable destinations by pointing them to a NULL interface instead of valid gateway address. Any traffic that has a destination matching this undesirable destination shall be dropped automatically in hardware without going to the CPU.

Use the **null** option in the **ip static-route** command to create an IPv4 blackhole route:

```
-> ip static-route 55.0.0.0/8 gateway null
```

Alternatively, the gateway address ‘0.0.0.0’ can be used to create an IPv4 blackhole route.

```
-> ip static-route 55.0.0.0/8 gateway 0.0.0.0
```

Configuring an IP Routed Port

An IP interface can be configured on a VLAN and a port or linkagg can be added to this VLAN as tagged or untagged, in a single CLI command. Use the **ip interface rtr-port** to create a VLAN, configure an IP interface on that VLAN and assign a single port as tagged or untagged to that VLAN.

For example.

- To create VLAN interface and assign port 1/1 as tagged port to that VLAN use the below command:

```
-> ip interface test address 10.0.0.1/8 vlan 30 rtr-port port 1/1 tagged
```

- To create VLAN interface and assign port 1/2 as untagged port to that VLAN use the below command:

```
-> ip interface test1 address 20.0.0.1/8 vlan 40 rtr-port port 1/2 untagged
```

Create a linkagg and then create a VLAN interface and assign the created linkagg as tagged or untagged to that VLAN.

For example.

- To create VLAN interface and assign linkagg 6 as tagged to that VLAN use the below command:

```
-> ip interface test address 10.0.0.1/8 vlan 30 rtr-port linkagg 6 tagged
```

- To create VLAN interface and assign linkagg 7 as untagged to that VLAN use the below command:

```
-> ip interface test1 address 20.0.0.1/8 vlan 40 rtr-port linkagg 7 untagged
```

The VLAN associated with the router-port must be a new, unused VLAN. This VLAN is a routing-only VLAN with a single port or trunk. Configuration to add additional members to this VLAN, or to delete this VLAN directly using **no vlan** command is rejected. This vlan can only be deleted by deleting the associated IP interface using the **no** form of the **ip interface** command.

If the IP interface is modified such that it's no longer bound to this VLAN, the corresponding VLAN is deleted.

Configuring Address Resolution Protocol (ARP)

To send packets on a locally connected network, the switch uses ARP to match the IP address of a device with its physical (MAC) address. To send a data packet to a device with which it has not previously communicated, the switch first broadcasts an ARP request packet. The ARP request packet requests the Ethernet hardware address corresponding to an Internet address. All hosts on the receiving Ethernet receive the ARP request, but only the host with the specified IP address responds. If present and functioning, the host with the specified IP address responds with an ARP reply packet containing its hardware address. The switch receives the ARP reply packet, stores the hardware address in its ARP cache for future use, and begins exchanging packets with the receiving device.

The switch stores the hardware address in its ARP cache (ARP table). The table contains a listing of IP addresses and their corresponding translations to MAC addresses. Entries in the table are used to translate 32-bit IP addresses into 48-bit Ethernet or IEEE 802.3 hardware addresses. Dynamic addresses remain in the table until they time out. You can set this time-out value and you can also manually add or delete permanent addresses to/from the table.

Adding a Permanent Entry to the ARP Table

As described above, dynamic entries remain in the ARP table for a specified time period before they are automatically removed. However, you can create a permanent entry in the table.

Use the **arp** command to add a permanent entry to the ARP table. Enter the IP address of the entry followed by its physical (MAC) address. For example, to create an entry for IP address 171.11.1.1 with a corresponding physical address of 00:05:02:c0:7f:11, enter the following command:

```
-> arp 171.11.1.1 00:05:02:c0:7f:11
```

Configuring a permanent ARP entry with a multicast address is also supported. For example, the following command creates a permanent multicast ARP entry:

```
-> arp 2.2.3.40 01:4a:22:03:44:5c
```

When configuring a static multicast ARP entry, do not use any of the following multicast addresses:

01:00:5E:00:00:00 to 01:00:5E:7F:FF:FF
01:80:C2:XX.XX.XX
33:33:XX:XX:XX:XX

The IP address and hardware address (MAC address) are *required* when you add an entry to the ARP table. Optionally, you can also specify:

- **Alias.** Use the **alias** keyword to specify that the switch acts as an alias (proxy) for this IP address. When the alias option is used, the switch responds to all ARP requests for the specified IP address with its own MAC address. This option is not related to Proxy ARP as defined in RFC 925. For example:

  ```
  -> arp 171.11.1.1 00:05:02:c0:7f:11 alias
  ```

- **ARP Name**. Specify a name for the permanent ARP entry. For example:

  ```
  -> arp 171.11.1.1 00:2a:90:d1:8e:10 server1
  ```

- **Interface**. Use the **interface** parameter to set the interface for the permanent ARP entry. For example:

  ```
  -> arp 171.11.1.1 00:2a:90:d1:8e:10 interface Int1 server1
  ```

Use the **show arp** command to display the ARP table.

Note. As most hosts support the use of address resolution protocols to determine and cache address information (called dynamic address resolution), it is not required to specify permanent ARP entries.

Deleting a Permanent Entry from the ARP Table

Permanent entries do not age out of the ARP table. Use the **no arp** command to delete a permanent entry from the ARP table. When deleting an ARP entry, you only need to enter the IP address. For example, to delete an entry for IP address 171.11.1.1, enter the following command:

```
-> no arp 171.11.1.1
```

Use the **show arp** command to display the ARP table and verify that the entry was deleted.

Note. You can also use the **no arp** command to delete a dynamic entry from the table.

Clearing a Dynamic Entry from the ARP Table

Dynamic entries can be cleared using the **clear arp-cache** command. This command clears all dynamic entries. Clear the permanent entries using the **no arp** command.

Use the **show arp** command to display the table and verify that the table was cleared.

Note. Dynamic entries remain in the ARP table until they time out. If the switch does not receive data from a host for this user-specified time, the entry is removed from the table. If another packet is received from this host, the switch goes through the discovery process again to add the entry to the table. The switch uses the MAC Address table time-out value as the ARP time-out value. Use the **mac-learning aging-time** command to set the time-out value.

Local Proxy ARP

The Local Proxy ARP feature is an extension of the Proxy ARP feature, but is enabled on an IP interface and applies to the VLAN bound to that interface. When Local Proxy ARP is enabled, all ARP requests received on VLAN member ports are answered with the MAC address of the IP interface that has Local Proxy ARP enabled. In essence, all VLAN traffic is now routed within the VLAN instead of bridged.

This feature is intended for use with port mapping applications where VLANs are one-port associations. This allows hosts on the port mapping device to communicate through the router. ARP packets are still bridged across multiple ports.

Local Proxy ARP takes precedence over any switch-wide Proxy ARP or ARP function. In addition, it is not necessary to configure Proxy ARP to use Local Proxy ARP. The two features are independent of each other.

By default, Local Proxy ARP is disabled when an IP interface is created. To enable this feature, use the **ip interface** command. For example:

```
-> ip interface Accounting local-proxy-arp
```

When Local Proxy ARP is enabled for any one IP router interface associated with a VLAN, the feature is applied to the entire VLAN. It is not necessary to enable it for each interface. However, if the IP interface that has the Local Proxy ARP feature enabled is moved to another VLAN, Local Proxy ARP is enabled for the new VLAN and must be enabled on another interface for the old VLAN.

Dynamic Proxy ARP - MAC Forced Forwarding

Dynamic Proxy ARP - MAC Forced Forwarding is used to forward all traffic from L2 clients to a head-end router. This head-end router filters and forwards the traffic from the local network or back to other clients in the same VLAN/IP subnet. In order to accomplish this, Dynamic Proxy ARP combines the functionality of other switch features to dynamically learn router addresses and act as a proxy for that router. Dynamic Proxy ARP - MAC Forced Forwarding uses the following features:

Port Mapping - Port Mapping forwards traffic from user-ports only to network-ports, preventing communication between L2 clients in the same VLAN. Port mapping prevents direct communication between clients in the same VLAN forcing all traffic to be forwarded to the head end router.

Proxy ARP - All ARP requests received on port mapping user-ports are answered with the MAC address of the head end router. Dynamic Proxy ARP dynamically learns the IP and MAC address of a head end router and responds with that router MAC address instead of flooding the ARP request.

DHCP Snooping - Snoops the DHCP packets between the server and clients. DHCP snooping is used to dynamically learn the IP address of the head end router.

MAC Forced Forwarding Steps:

1. Clients are connected to the user-ports of a port mapping session.

2. Head end router is connected to the network-port of the same port mapping session.

3. DHCP snooping is enabled and uses the DHCP DISCOVER and DHCP ACK packets to learn the head end router IP.

4. The ARP Request and Reply is snooped by the switch to learn the head end router MAC address.

5. The ARP Requests from clients on the user-ports are intercepted by the switch and the switch replies with the head end router MAC address.

6. All traffic from the clients is now forwarded to the head end router to be filtered.

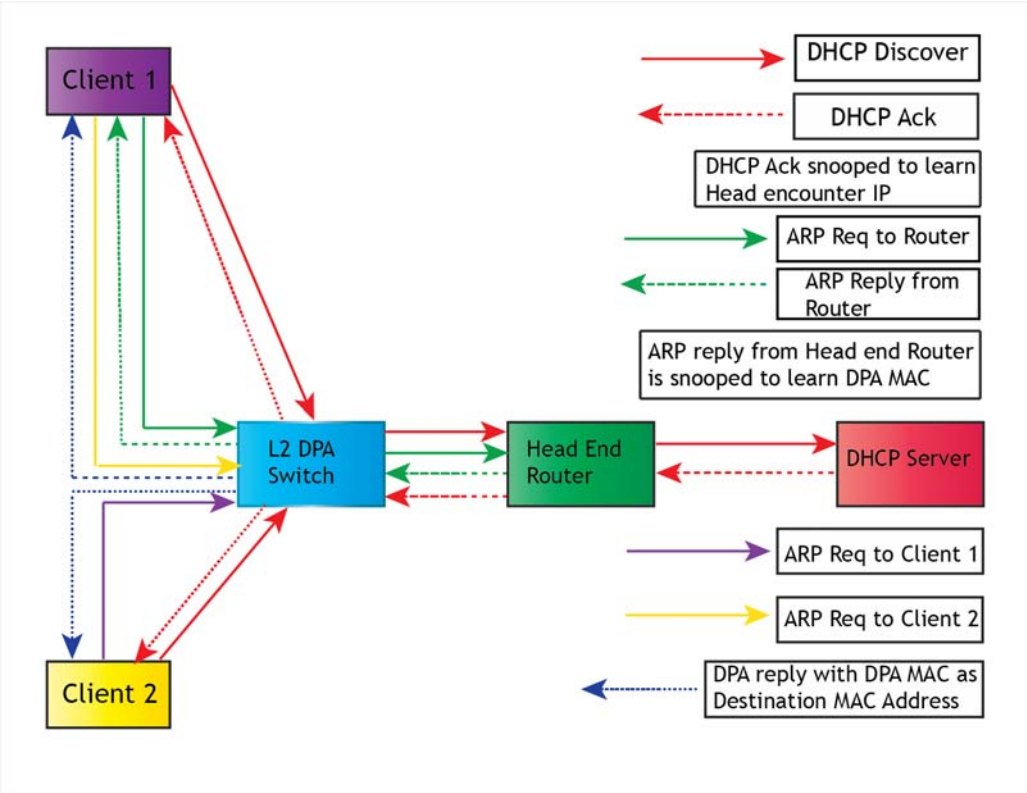


Figure 17-2 : Dynamic Proxy ARP

Use the **port-mapping user-port network-port** and **dhcp-snooping vlan** commands as follows to enable Dynamic Proxy ARP - MAC Forced Forwarding. For example:

```
-> port mapping 1 user-port 1/1/1-2 network-ports 1/1/3
-> port mapping 1 dynamic-proxy-arp enable
-> dhcp-snooping vlan 1
```

The example above considers that all devices are in VLAN 1, Clients 1 and 2 are connected to ports 1/1/1 and 1/1/2, and the head end router is connected to port 1/1/3.

ARP Filtering

ARP filtering is used to determine whether the switch responds to ARP requests that contain a specific IP address. ARP filtering is used in conjunction with the Local Proxy ARP application; however, it is available for use on its own or with other applications.

By default, no ARP filters exist in the switch configuration. When there are no filters present, all ARP packets are processed, unless they are blocked or redirected by some other feature.

Use the **arp filter** command to specify the following parameter values required to create an ARP filter:

- An IP address (for example, 193.204.173.21) used to determine whether an ARP packet is filtered.
- An IP mask (for example, 255.0.0.0) used to identify which part of the ARP packet IP address is compared to the filter IP address.
- An optional VLAN ID to specify that the filter is only applied to ARP packets from that VLAN.
- Which ARP packet IP address to use for filtering (sender or target). If the target IP address in the ARP packet matches a target IP specified in a filter, then the disposition for that filter applies to the ARP packet. If the sender IP address in the ARP packet matches a sender IP specified in a filter, then the disposition for that filter applies to the ARP packet.
- The filter disposition (block or allow). If an ARP packet meets filter criteria, the switch is either blocked from responding to the packet or allowed to respond to the packet depending on the filter disposition. Packets that do not meet any filter criteria are responded to by the switch.

The following **arp filter** command example creates an ARP filter, which blocks the switch from responding to ARP packets that contain a sender IP address that starts with 198:

```
-> arp filter 198.0.0.0 mask 255.0.0.0 sender block
```

Up to 200 ARP filters can be defined on a single switch. To remove an individual filter, use the no form of the **arp filter** command. For example:

```
-> no arp filter 198.0.0.0
```

To clear all ARP filters from the switch configuration, use the **clear arp filter** command. For example:

```
-> clear arp filter
```

Use the **show arp filter** command to verify the ARP filter configuration. For more information on ARP Filtering and other ARP filter commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Gratuitous ARP

A Gratuitous ARP is an ARP broadcast in which the source and destination MAC addresses are the same. It is used to inform the network about a host IP address. A spoofed Gratuitous ARP message can cause network mapping information to be stored incorrectly, causing network malfunction. The OmniSwitch allows to configure the Gratuitous ARP.

The incoming and outgoing Gratuitous ARP packets can be enabled or disabled on the switch.

Outgoing Gratuitous ARP Configuration

By default, the outgoing Gratuitous ARP packets are enabled. The switch will send Gratuitous ARP packets every five minutes. To filter the Gratuitous packet, it must be disabled. The outgoing Gratuitous

ARP packets can be enabled or disabled using the **arp send-gratuitous-arp** CLI command. For example, to disable:

```
-> ip send-gratuitous-arp disable
```

For example, to enable:

```
-> ip send-gratuitous-arp enable
```

Note. The configuration status of the outgoing Gratuitous ARP packets can be viewed using the **show arp send-gratuitous-arp** CLI command.

Incoming Gratuitous ARP Configuration

By default, the incoming Gratuitous ARP packets are not blocked. To block the incoming Gratuitous ARP packets the feature must be enabled. The incoming Gratuitous ARP packets can be configured using the **ip dos type** CLI command.

For example, to block the incoming Gratuitous ARP packets:

```
-> ip dos type gratuitous-arp admin-state enable
```

Note. The configuration status of the incoming Gratuitous ARP packets cab be viewed using the **show ip dos config** CLI command.

IP Configuration

IP is enabled on the switch by default and a few options that can, or need to be, configured. This section provides instructions for basic IP configuration options.

Configuring the Router Primary Address

By default, the router primary address is derived from the first IP interface that becomes operational on the router. if the router ID is not a valid IP unicast address, the router primary IP address is used by BGP to derive its unique BGP Identifier.

Use the **ip router primary-address** command to configure the router primary address. Enter the command, followed by the IP address. For example, to configure a router primary address of 172.22.2.115, you must enter:

```
-> ip router primary-address 172.22.2.115
```

Configuring the Router ID

By default, the router primary address of the router is used as the router ID. However, if a primary address has not been explicitly configured, the router ID defaults to the address of the first IP interface that becomes operational.

Use the **ip router router-id** command to configure the router ID. Enter the command, followed by the IP address. For example, to configure a router ID of 172.22.2.115, you would enter:

```
-> ip router router-id 172.22.2.115
```

There is no CLI command to remove a configured router ID. In order to remove the router ID, it must be set back to the default value by performing the following:

1. Take the last 4 octets of the chassis MAC address, use the **show chassis** command.
2. Convert them to a decimal IPv4 address.
3. Set the router ID to that address.
4. The router ID will no longer show up in the configuration snapshot.

For example, if the chassis MAC address is 00:e0:b1:28:1c:89, the last 4 octets are b1:28:1c:89. Converting to decimal results in the IP address of 177.40.28.137. Configure the router ID to that IP address, for example:

```
-> ip router router-id 177.40.28.137
```

The router ID will now no longer be displayed in the configuration snapshot.

Configuring the Route Preference of a Router

By default, the route preference of a router is in this order: local, static, OSPF, RIP, EBGP, and IBGP (highest to lowest).

Use the **ip route-pref** command to change the route preference value of a router. For example, to configure the route preference of an OSPF route, you must enter:

```
-> ip route-pref ospf 15
```

To display the current route preference configuration, use the **show ip route-pref** command:

```
-> show ip route-pref
  Protocol    Route Preference Value
------------+------------------------
  Local                 1
  Static                2
  OSPF                110
  RIP                 120
  EBGP                190
  IBGP                200
```

Configuring the Time-to-Live (TTL) Value

The TTL value is the default value inserted into the TTL field of the IP header of datagrams originating from the switch whenever a TTL value is not supplied by the transport layer protocol. The value is measured in hops.

Use the **ip default-ttl** command to set the TTL value. Enter the command, followed by the TTL value. For example, to set a TTL value of 75, you would enter:

```
-> ip default-ttl 75
```

The default hop count is 64. The valid range is 1 to 255. Use the **show ip config** command to display the default TTL value.

Configuring Route Map Redistribution

You can learn and advertise IPv4 routes between different protocols. Such a process is referred to as route redistribution and is configured using the **ip redist** command.

Redistribution uses route maps to control how external routes are learned and distributed. A route map consists of one or more user-defined statements that can determine which routes are allowed or denied access to the receiving network. In addition, a route map can also contain statements that modify route parameters before they are redistributed.

When a route map is created, a name is given to identify the group of statements that it represents. This name is required by the **ip redist** command. Therefore, configuring route redistribution involves the following steps:

1 Create a route map, as described in "Using Route Maps" on page 17-20.

2 Configure redistribution to apply a route map, as described in "Configuring Route Map Redistribution" on page 17-24.

Using Route Maps

A route map specifies the criteria that are used to control redistribution of routes between protocols. Such criteria is defined by configuring route map statements. There are three different types of statements:

- **Action**. An action statement configures the route map name, sequence number, and whether redistribution is permitted or denied based on route map criteria.
- **Match.** A match statement specifies criteria that a route must match. When a match occurs, then the action statement is applied to the route.

- **Set.** A set statement is used to modify route information before the route is redistributed into the receiving protocol. This statement is only applied if all the criteria of the route map is met and the action permits redistribution.

The **ip route-map** command is used to configure route map statements and provides the following **action**, **match**, and **set** parameters:

ip route-map action ...	ip route-map match ...	ip route-map set ...
permit **deny**	**ip-address** **ip-nexthop** **ipv6-address** **ipv6-nexthop** **tag** **ipv4-interface** **ipv6-interface** **metric** **route-type** **protocol** **name**	**metric** **metric-type** **tag** **community** **local-preference** **level** **ip-nexthop** **ipv6-nexthop**

Refer to the "IP Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the **ip route-map** command parameters and usage guidelines.

Once a route map is created, it is then applied using the **ip redist** command. See "Configuring Route Map Redistribution" on page 17-24 for more information.

Route Maps are also used for VRF route leaking and RIP route filtering. See "VRF Route Leak" on page 17-41 section for more information.

Creating a Route Map

When a route map is created, a name (up to 20 characters), a sequence number, and an action (permit or deny) is specified. Specifying a sequence number is optional. If a value is not configured, then the number 50 is used by default.

To create a route map, use the **ip route-map** command with the **action** parameter. For example,

```
-> ip route-map ospf-to-bgp sequence-number 10 action permit
```

The above command creates the ospf-to-bgp route map, assigns a **sequence number** of 10 to the route map, and specifies a **permit** action.

To optionally filter routes before redistribution, use the **ip route-map** command with a **match** parameter to configure match criteria for incoming routes. For example,

```
-> ip route-map ospf-to-bgp sequence-number 10 match tag 8
```

The above command configures a match statement for the ospf-to-bgp route map to filter routes based on their tag value. When this route map is applied, only OSPF routes with a tag value of eight are redistributed into the BGP network. All other routes with a different tag value are dropped.

Note. Configuring match statements is not required. However, if a route map does not contain any match statements and the route map is applied using the **ip redist** command, the router redistributes *all* routes into the network of the receiving protocol.

Use the **ip route-map** command with a **set** parameter to modify route information before redistribution. For example,

```
-> ip route-map ospf-to-bgp sequence-number 10 set tag 5
```

The above command configures a set statement for the ospf-to-bgp route map that changes the route tag value to five. As this statement is part of the ospf-to-bgp route map, it is only applied to routes that have an existing tag value equal to eight.

The following is a summary of the commands used in the above examples:

```
-> ip route-map ospf-to-bgp sequence-number 10 action permit
-> ip route-map ospf-to-bgp sequence-number 10 match tag 8
-> ip route-map ospf-to-bgp sequence-number 10 set tag 5
```

To verify a route map configuration, use the **show ip route-map** command:

```
-> show ip route-map
Route Maps: configured: 1 max: 200
Route Map: ospf-to-bgp Sequence Number: 10 Action permit
  match tag 8
  set tag 5
```

Deleting a Route Map

Use the **no** form of the **ip route-map** command to delete an entire route map, a route map sequence, or a specific statement within a sequence.

To delete an entire route map, enter **no ip route-map** followed by the route map name. For example, the following command deletes the entire route map named redistipv4:

```
-> no ip route-map redistipv4
```

To delete a specific sequence number within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the actual number. For example, the following command deletes sequence 10 from the redistipv4 route map:

```
-> no ip route-map redistipv4 sequence-number 10
```

In the above example, the redistripv4 route map is not deleted. Only those statements associated with sequence 10 are removed from the route map.

To delete a specific statement within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the sequence number for the statement, then either **match** or **set** and the match or set parameter and value. For example, the following command deletes only the match tag 8 statement from route map redistipv4 sequence 10:

```
-> no ip route-map redistipv4 sequence-number 10 match tag 8
```

Configuring Route Map Sequences

A route map can consist of one or more sequences of statements. The sequence number determines which statements belong to which sequence and the order in which sequences for the same route map are processed.

To add match and set statements to an existing route map sequence, specify the same route map name and sequence number for each statement. For example, the following series of commands creates route map rm_1 and configures match and set statements for the rm_1 sequence 10:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 set metric 1
```

To configure a new sequence of statements for an existing route map, specify the same route map name but use a different sequence number. For example, the following commands create a sequence 20 for the rm_1 route map:

```
-> ip route-map rm_1 sequence-number 20 action permit
-> ip route-map rm_1 sequence-number 20 match ipv4-interface to-finance
-> ip route-map rm_1 sequence-number 20 set metric 5
```

The resulting route map appears as follows:

```
-> show ip route-map rm_1
Route Map: rm_1 Sequence Number: 10 Action permit
  match tag 8
  set metric 1
Route Map: rm_1 Sequence Number: 20 Action permit
  match ip4 interface to-finance
  set metric 5
```

Sequence 10 and sequence 20 are both linked to route map rm_1 and are processed in ascending order according to their sequence number value. There is an implied logical OR between sequences. As a result, if there is no match for the tag value in sequence 10, then the match interface statement in sequence 20 is processed. However, if a route matches the tag 8 value, then sequence 20 is not used. The set statement for whichever sequence was matched is applied.

A route map sequence can contain multiple match statements. If these statements are of the same kind (for example, match tag 5, match tag 8, and so on) then a logical OR is implied between each like statement. If the match statements specify different types of matches (for example, match tag 5, match ip4 interface to-finance, and so on), then a logical AND is implied between each statement. For example, the following route map sequence redistributes a route if its tag is either 8 or 5:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
```

If the route has a tag of 8 or 5 and the route was learned on the IPv4 interface to-finance, the following route map sequence redistributes a route:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 match ipv4-interface to-finance
```

Configuring Access Lists

An IP access list provides a convenient way to add multiple IPv4 or IPv6 addresses to a route map. Using an access list avoids having to enter a separate route map statement for each individual IP address. Instead, a single statement is used that specifies the access list name. The route map is then applied to all the addresses contained within the access list.

Configuring an IP access list involves two steps: creating the access list and adding IP addresses to the list. To create an IP access list, use the **ip access-list** command (IPv4) or the **ipv6 access-list** command (IPv6) and specify a name to associate with the list. For example,

```
-> ip access-list ipaddr
-> ipv6 access-list ip6addr
```

To add addresses to an access list, use the **ip access-list address** (IPv4) or the **ipv6 access-list address** (IPv6) command. For example, the following commands add addresses to an existing access list:

```
-> ip access-list ipaddr address 10.0.0.0/8
-> ipv6 access-list ip6addr address 2001::/64
```

Use the same access list name each time the above commands are used to add additional addresses to the same access list. In addition, both commands provide the ability to configure if an address and/or its matching subnet routes are permitted (the default) or denied redistribution. For example:

```
-> ip access-list ipaddr address 16.24.2.1/16 action deny redist-control all-
subnets
-> ipv6 access-list ip6addr address 2001::1/64 action permit redist-control no-
subnets
```

For more information about configuring access list commands, see the "IP Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Route Map Redistribution

The **ip redist** command is used to configure the redistribution of routes from a source protocol into the destination protocol. This command is used on the IPv4 router that performs the redistribution.

A source protocol is a protocol from which the routes are learned. A destination protocol is the one into which the routes are redistributed. Ensure that both protocols are loaded and enabled before configuring redistribution.

Redistribution applies criteria specified in a route map to routes received from the source protocol. Therefore, configuring redistribution requires an existing route map. For example, the following command configures the redistribution of OSPF routes into a BGP network using the ospf-to-bgp route map:

```
-> ip redist ospf into bgp route-map ospf-to-bgp
```

OSPF routes received by the router interface are processed based on the contents of the ospf-to-bgp route map. Routes that match criteria specified in this route map are either allowed or denied redistribution into the BGP network. The route map can also specify the modification of route information before the route is redistributed. See "Using Route Maps" on page 17-20 for more information.

To remove a route map redistribution configuration, use the **no** form of the **ip redist** command. For example:

```
-> no ip redist ospf into bgp route-map ospf-to-bgp
```

Use the **show ip redist** command to verify the redistribution configuration:

```
-> show ip redist

Source       Destination
Protocol     Protocol     Status    Route Map
------------+------------+---------+--------------------
LOCAL4       RIP          Enabled   rip_1
LOCAL4       OSPF         Enabled   ospf_2
LOCAL4       BGP          Enabled   bgp_3
RIP          OSPF         Enabled   ospf-to-bgp
```

Configuring the Administrative Status of the Route Map Redistribution

The administrative status of a route map redistribution configuration is enabled by default. To change the administrative status, use the **status** parameter with the **ip redist** command. For example, the following command disables the redistribution administrative status for the specified route map:

```
-> ip redist ospf into bgp route-map ospf-to-bgp status disable
```

The following command example enables the administrative status:

```
-> ip redist ospf into bgp route-map ospf-to-bgp status enable
```

Route Map Redistribution Example

The following example configures the redistribution of OSPF routes into a BGP network using a route map (ospf-to-bgp) to filter specific routes:

```
-> ip route-map ospf-to-bgp sequence-number 10 action deny
-> ip route-map ospf-to-bgp sequence-number 10 match tag 5
-> ip route-map ospf-to-bgp sequence-number 10 match route-type external type2

-> ip route-map ospf-to-bgp sequence-number 20 action permit
-> ip route-map ospf-to-bgp sequence-number 20 match ipv4-interface intf_ospf
-> ip route-map ospf-to-bgp sequence-number 20 set metric 255

-> ip route-map ospf-to-bgp sequence-number 30 action permit
-> ip route-map ospf-to-bgp sequence-number 30 set tag 8

-> ip redist ospf into bgp route-map ospf-to-bgp
```

The resulting ospf-to-bgp route map redistribution configuration does the following

- Denies the redistribution of Type 2 external OSPF routes with a tag set to five.
- Redistributes into BGP all routes learned on the intf_ospf interface and sets the metric for such routes to 255.
- Redistributes into BGP all other routes that are not processed by sequence 10 or 20, and sets the tag for such routes to eight.

Route Map Match/Set Validity Table

Different applications support different match/set types. Since a route-map can be in use by all applications at the same time a route-map modification may be valid for one application but illegal for another. The modification will fail if it is illegal for one or more in use applications. If a route-map is in-use by SPB it cannot be modified. The table below displays the validity per application for each match/set entity.

Match/Set	**Redist IPv4**	**Redist IPv6**	**Export IPv4**	**Export IPv6**	**IPv4 Import**	**Import IPv6**	**RIP Filter**
Match ip-address		E		E	1E	E	1E
Match-ip-address (access-list)		E		E	1E	E	1E
Match ip-nexthop		E		E		E	E
Match ip-nexthop (access-list)	7E	E	7E	E	7E	E	E
Match ipv6-address	E		E		E	1E	E
Match ipv6-address (access-list)	E		E		E	1E	E
Match ipv6-nexthop	E		E		E		E
Match ipv6-nexthop (access-list)	E	7E	E	7E	E	7E	E
Match tag							E
Match metric							E

Match/Set	Redist IPv4	Redist IPv6	Export IPv4	Export IPv6	IPv4 Import	Import IPv6	RIP Filter
Match route-type	2E	2E			E	E	E
Match ipv4-interface		E		E	E	E	E
Match ipv6-interface	E		E		E	E	E
Match protocol	E	E			E	E	E
Match name	3E	3E			E	E	E
Set metric							E
Set metric-type	4E	4E	E	E	E	E	E
Set level	5E	5E	E	E	E	E	E
Set tag							E
Set community	6E	6E	E	E	E	E	E
Set local-pref	6E	6E	E	E	E	E	E
Set ip-nexthop		E	E	E	E	E	E
Set ipv6-nexthop	E		E	E	E	E	E

E: Error
1: Error if aggregate
2: Internal for source RIP; Internal/iExternal/External Type1/External Type2 for source OSPF; Internal/External for source BGP; L1/L2 for source ISIS; ISIS into ISIS L1->L2 or L2->L1 only
3: Ignore if not a static route
4: Internal/External Type1/External Type2 for destination OSPF; Internal/External for destination BGP
5: L1/L2 for destination ISIS; ISIS into ISIS L1->L2 or L2->L1 only
6: Error if not BGP destination
7: Error if aggregate or no-subnet

IP-Directed Broadcasts

An IP directed broadcast is an IP datagram that has all zeros or all 1 in the host portion of the destination IP address. The packet is sent to the broadcast address of a subnet to which the sender is not directly attached. Directed broadcasts are used in denial-of-service attacks. In a denial-of-service attack, a continuous stream of ping requests is sent from a falsified source address to a directed broadcast address, resulting in a large stream of replies, which can overload the host of the source address. By default, the switch drops directed broadcasts. Directed broadcasts must not be enabled.

Use the **ip directed-broadcast** command to enable or disable IP-directed broadcasts. For example:

```
-> ip directed-broadcast enable
-> ip directed-broadcast disable
```

When IP directed broadcast is enabled, by default, it is enabled on the 'default' VRF.

Controlled Directed Broadcasts

The Control Directed Broadcast can be configured to direct only the packet from trusted source to the destined network, while the other directed broadcast packets are dropped.

To support the control directed broadcast, specify the source IP address, destination IP address and destination VLAN information to broadcast the packets in controlled manner. The specified information is considered as the trusted information to broadcast the packets received from the defined parameters, and the remaining broadcast packets are dropped.

IP directed broadcast must be enabled for the controlled IP directed broadcast to work.

When controlled IP directed broadcast is enabled, by default, it is enabled on the 'default' VRF. The trusted information must have the source IP with optional destination IP address or VLAN ID.

For example:

```
-> ip directed-broadcast trusted-source-ip 30.0.0.0 mask 255.255.255.0

-> vrf test123 ip directed-broadcast trusted-source-ip 30.0.0.0 mask
255.255.255.0 destination-ip 10.0.0.255 destination-mask 255.255.255.255

-> ip directed-broadcast trusted-source-ip 30.0.0.0/24 destination-ip
10.0.0.255/24

-> ip directed-broadcast trusted-source-ip 30.0.0.0 mask 255.255.255.0 destina-
tion-vlan 10

-> ip directed-broadcast trusted-source-ip 30.0.0.0/24 destination-vlan 10-15
```

Use the **show ip directed-broadcast** command to display the configured trusted source IP address. Use **details** keyword in this command to view the destination IP addresses or VLANs information for the specified source IP.

Denial of Service (DoS) Filtering

By default, the switch filters denial of service (DoS) attacks, which are security attacks aimed at devices that are available on a private network or the Internet. Some attacks aim at system bugs or vulnerability, while other types of attacks involve generating large volumes of traffic so that network service is denied to legitimate network users. These attacks include the following:

- ICMP Ping of Death—Ping packets that exceed the largest IP datagram size (65535 bytes) are sent to a host and crash the system.
- Land Attack—Spoofed packets are sent with the SYN flag set to a host on any open port that is listening. The machine can crash or reboot in an attempt to respond.
- ARP Flood Attack—Floods a switch with a large number of ARP requests, resulting in the switch using a large amount of the CPU time to respond to these requests. If the number of ARP requests exceeds the preset value of 500 per second, an attack is detected.
- Invalid IP Attack—Packets with invalid source or destination IP addresses are received by the switch. When such an Invalid-IP attack is detected, the packets are dropped, and SNMP traps are generated. Following are few examples of invalid source and destination IP addresses:

Invalid Source IP address	• 0.x.x.x. • 255.255.255.255. • subnet broadcast, that is, 172.28.255.255, for an existing IP interface 172.28.0.0/16. • in the range 224.x.x.x - 255.255.255.254. • Source IP address equals one of Switch IP Interface addresses.
Invalid Destination IP address	• 127.x.x.x. • in the range 240.x.x.x - 255.255.255.254. • 0.0.0.0 (valid exceptions- certain DHCP packets). • 172.28.0.0 for a router network 172.28.4.11/16. • 0.x.x.x.

- Multicast IP and MAC Address Mismatch—This attack is detected when:
  - the source MAC address of a packet received by a switch is a Multicast MAC address.
  - the destination IP and MAC addresses of a packet received by a switch is same as the Multicast IP and MAC addresses, but the Multicast IP and the Multicast MAC addresses do not match.

Note. In both the conditions described above in “Multicast IP and MAC Address Mismatch”, packets are dropped and SNMP traps are generated.

  - the destination IP is a unicast IP and the destination MAC address is either a Broadcast or Multicast address. In such a condition, an event is recorded in the DoS statistics. No SNMP traps are generated as valid packets can also fall under this category.

- Ping overload—Floods a switch with a large number of ICMP packets, resulting in the switch using a large amount of CPU time to respond to these packets. If the number of ICMP packets exceed 100 per second, a DoS attack is detected. By default, the detection of attack is disabled.

- Packets with loopback source IP address—Packets with an invalid source address of 127.0.0.0/8 (loopack network) are received by the switch. When such packets are detected, they are dropped, and SNMP traps are generated.

- IP options filter—The ingress packets with IP options are detected and dropped.

The switch can be set to detect various types of port scans by monitoring for TCP or UDP packets sent to open or closed ports. Monitoring is done in the following manner:

- **Packet penalty values set**. TCP and UDP packets destined for open or closed ports are assigned a penalty value. Each time a packet of this type is received, its assigned penalty value is added to a running total. This total is cumulative and includes all TCP and UDP packets destined for open or closed ports.

- **Port scan penalty value threshold**. The switch is given a port scan penalty value threshold. This number is the maximum value the running penalty total can achieve before triggering an SNMP trap.

- **Decay value**. A decay value is set. The running penalty total is divided by the decay value every minute.

- **Trap generation**. If the total penalty value exceeds the set port scan penalty value threshold, a trap is generated to alert the administrator that a port scan can be in progress.

For example, imagine that a switch is set so that TCP and UDP packets destined for closed ports are given a penalty of 10, TCP packets destined for open ports are given a penalty of 5, and UDP packets destined for open ports are given a penalty of 20. The decay is set to 2, and the switch port scan penalty value threshold is set to 2000:.

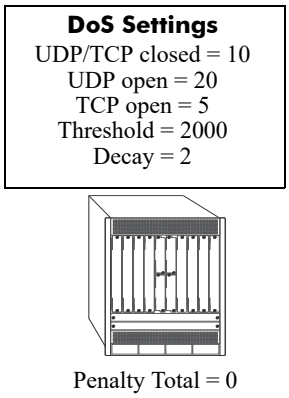


Figure 17-3 : Denial of Service (DoS) Filtering-1

In 1 minute, 10 TCP closed port packets and 10 UDP closed port packets are received. This brings the total penalty value to 200, as shown using the following equation:

```
(10 TCP X 10 penalty) + (10 UDP X 10 penalty) = 200
```

This value would be divided by 2 (due to the decay) and decreased to 100. The switch would not record a port scan:

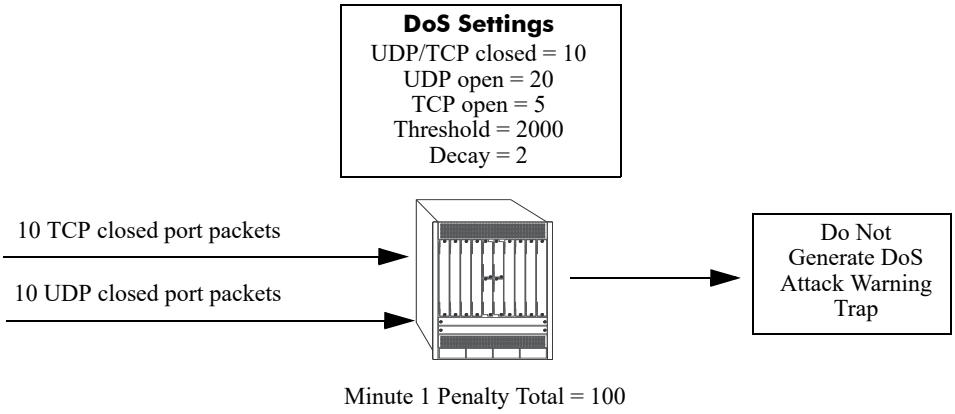


Figure 17-4 : Denial of Service (DoS) Filtering-2

In the next minute, 10 more TCP and UDP closed port packets are received, along with 200 UDP open port packets. This would bring the total penalty value to 4300, as shown using the following equation:

```
(100 previous minute value) + (10 TCP X 10 penalty) + (10 UDP X 10 penalty) +
(200 UDP X 20 penalty) = 4300
```

This value would be divided by 2 (due to decay) and decreased to 2150. The switch would record a port scan and generate a trap to warn the administrator:

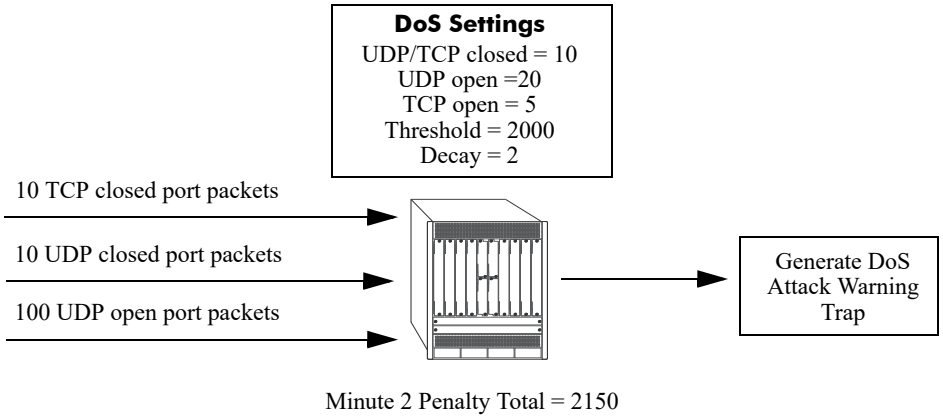


Figure 17-5 : Denial of Service (DoS) Filtering-3

The above functions and how to set their values are covered in the sections that follow.

Setting Penalty Values

You can set a penalty value for the following types of traffic:

- TCP/UDP packets bound for closed ports.
- TCP traffic bound for open ports.
- UDP traffic bound for open ports.

Each type has its own command to assign a penalty value. Penalty values can be any non-negative integer. Each time a packet is received that matches an assigned penalty, the total penalty value for the switch is increased by the penalty value of the packet in question.

To assign a penalty value to TCP/UDP packets bound for a closed port, use the **ip dos scan close-port-penalty** command with a penalty value. For example, to assign a penalty value of 10 to TCP/UDP packets destined for closed ports, enter the following:

```
-> ip dos scan close-port-penalty 10
```

To assign a penalty value to TCP packets bound for an open port, use the **ip dos scan tcp open-port-penalty** command with a penalty value. For example, to assign a penalty value of 10 to TCP packets destined for opened ports, enter the following:

```
-> ip dos scan tcp open-port-penalty 10
```

To assign a penalty value to UDP packets bound for an open port, use the **ip dos scan udp open-port-penalty** command with a penalty value. For example, to assign a penalty value of 10 to TCP/UDP packets destined for closed ports, enter the following:

```
-> ip dos scan udp open-port-penalty 10
```

Setting the Port Scan Penalty Value Threshold

The port scan penalty value threshold is the highest point the total penalty value for the switch can reach before a trap is generated informing the administrator that a port scan is in progress.

To set the port scan penalty value threshold, enter the threshold value with the **ip dos scan threshold** command. For example, to set the port scan penalty value threshold to 2000, enter the following:

```
-> ip dos scan threshold 2000
```

Setting the Decay Value

The decay value is the amount the total penalty value is divided by every minute. As the switch records incoming UDP and TCP packets, it adds their assigned penalty values together to create the total penalty value for the switch. To prevent the switch from registering a port scan from normal traffic, the decay value is set to lower the total penalty value every minute to compensate from normal traffic flow.

To set the decay value, enter the decay value with the **ip dos scan decay** command. For example, to set the decay value to 2, enter the following:

```
-> ip dos scan decay 2
```

Enabling DoS Traps

Enable the DoS traps for the switch to warn the administrator that a port scan can be in progress when the total penalty value of the switch crosses the port scan penalty value threshold.

To enable SNMP trap generation, enter the **ip dos trap** command, as shown:

```
-> ip dos trap enable
```

To disable DoS traps, enter the same **ip dos trap** command, as shown:

```
-> ip dos trap disable
```

ARP Poisoning

ARP Poisoning allows an attacker to sniff and tamper the data frames on a network. It also modifies or halts the traffic. The principle of ARP Poisoning is to send false or spoofed ARP messages to an Ethernet LAN.

The OmniSwitch introduces the functionality that detects the presence of an ARP poisoning host on a network. This functionality uses a configured restricted IP addresses, so that the switch does not get ARP response on sending an ARP request. If an ARP response is received, then an event is logged and the user is alerted using an SNMP trap.

Use the **ip dos arp-poison restricted-address** command to add an ARP Poison restricted address. Enter the command, followed by the IP address. For example, to add an ARP Poison restricted address as 192.168.1.1, you would enter:

```
-> ip dos arp-poison restricted-address 192.168.1.1
```

To delete an ARP Poison restricted address, enter **no ip dos arp-poison restricted-address** followed by the IP address. For example:

```
-> no ip dos arp-poison restricted-address 192.168.1.1
```

To verify the number of attacks detected for configured ARP poison restricted addresses, use the **show ip dos arp-poison** command. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Enabling/Disabling IP Services

When a switch initially boots up, all supported TCP/UDP well-known service ports are enabled (open). Although these ports provide access for essential switch management services, such as telnet, FTP, SNMP, they also are vulnerable to DoS attacks. It is possible to scan open service ports and launch such attacks based on well-known port information.

The **ip service** command allows you to disable (close) TCP/UDP well-known service ports selectively and enable them when necessary. This command only operates on TCP/UDP ports that are opened by default. It has no impact on ports that are opened by loading applications, such as RIP and BGP.

In addition, the **ip service** command allows you to designate which service to enable or disable by specifying the name of a service as well as changing the well-known port number associated with that service. For example, the following commands disable the telnet service, change the port and re-enable the service:

```
-> ip service telnet admin-state disable
-> ip service telnet port 20999
-> ip service telnet admin-state enable
```

Use **default** parameter to revert the port number of a service to the default port number.

```
-> ip service telnet port default
```

The following table lists **ip service** command options for specifying TCP/UDP services and also includes the well-known port number associated with each service:

service	port
ftp	21
ssh	22
telnet	23
http	80
https	443
ntp	123
snmp	161

Note: The NTP client functionality is disabled by default.

Managing IP

The following sections describe IP commands that can be used to monitor and troubleshoot IP forwarding on the switch.

Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP) is a network layer protocol within the IP protocol suite that provides message packets to report errors and other IP packet processing information back to the source. ICMP generates various kinds of useful messages, including Destination Unreachable, Echo Request and Reply, Redirect, Time Exceeded, and Router Advertisement and Solicitation. If an ICMP message cannot be delivered, a second one is not generated thus preventing an endless flood of ICMP messages.

When an ICMP destination-unreachable message is sent by a switch, it means that the switch is unable to send the package to its final destination. The switch then discards the original packet. There are two reasons why a destination is not reachable. Most commonly, the source host has specified a non-existent address. Less frequently, the switch does not have a route to the destination. The destination-unreachable messages include four basic types:

- Network-Unreachable Message—Usually means that a failure has occurred in the route lookup of the destination IP in the packet.
- Host-Unreachable Message—Usually indicates delivery failure, such as an unresolved client's hardware address or an incorrect subnet mask.
- Protocol-Unreachable Message—Usually means that the destination does not support the upper-layer protocol specified in the packet.
- Port-Unreachable Message—Implies that the TCP/UDP socket or port is not available.

Additional ICMP messages include:

- Echo-Request Message—Generated by the ping command, the message is sent by any host to test node reachability across an internetwork. The ICMP echo-reply message indicates that the node can be successfully reached.
- Redirect Message—Sent by the switch to the source host to stimulate more efficient routing. The switch still forwards the original packet to the destination. ICMP redirect messages allow host routing tables to remain small because it is necessary to know the address of only one switch, even if that switch does not provide the best path. Even after receiving an ICMP redirect message, few devices continue using the less-efficient route.
- Time-Exceeded Message—Sent by the switch if an IP packet’s TTL field reaches zero. If the internetwork contains a routing loop, the TTL field prevents packets from continuously circulating the internetwork. Once a packet TTL field reaches 0, the switch discards the packet.

Activating ICMP Control Messages

ICMP messages are identified by a *type* and a *code*. This number pair specifies an ICMP message. For example, ICMP type 4, code 0, specifies the source quench ICMP message.

To enable or disable an ICMP message, use the **icmp type** command with the type and code. For example, to enable the source quench the ICMP message (type 4, code 0) enter the following:

```
-> icmp type 4 code 0 enable
```

To list the ICMP message information use the **show icmp control** command.

In addition to the **icmp type** command, many commonly used ICMP messages have separate CLI commands for convenience. The following table lists the ICMP message name, type, and code:

ICMP Message	Command
Network unreachable (type 0, code 3)	**icmp unreachable**
Host unreachable (type 3, code 1)	**icmp unreachable**
Protocol unreachable (type 3, code 2)	**icmp unreachable**
Port unreachable (type 3, code 3)	**icmp unreachable**
Echo reply (type 0, code 0)	**icmp echo**
Echo request (type 8, code 0)	**icmp echo**
Timestamp request (type 13, code 0)	**icmp timestamp**
Timestamp reply (type 14, code 0)	**icmp timestamp**
Address Mask request (type 17, code 0)	**icmp addr-mask**
Address Mask reply (type 18, code 0)	**icmp addr-mask**

These commands are entered as the **icmp type** command, only without specifying a type or code. The echo, timestamp, and address mask commands have options for distinguishing between a request or a reply, and the unreachable command has options distinguishing between a network, host, protocol, or port.

For example, to enable an echo request message, enter the following:

```
-> icmp echo request enable
```

To enable a network unreachable message, enter the following:

```
-> icmp unreachable net-unreachable enable
```

Note. Enabling **host-unreachable** and **net-unreachable** messages are not recommended as it can cause the switch instability due to high-CPU conditions depending upon the volume of traffic required by these messages.

See Chapter 20, “IP Commands,” for specifics on the ICMP message commands.

Enabling All ICMP Types

To enable all ICMP message types, use the **icmp messages** command with the **enable** keyword. For example:

```
-> icmp messages enable
```

To disable all ICMP messages, enter the same command with the **disable** keyword. For example:

```
-> icmp messages enable
```

Setting the Minimum Packet Gap

The minimum packet gap is the time required between sending messages of a like type. For instance, if the minimum packet gap for Address Mask request messages is 40 microseconds, and an Address Mask message is sent, at least 40 microseconds must pass before another one could be sent.

To set the minimum packet gap, use the **min-pkt-gap** keyword with any of the ICMP control commands. For example, to set the Source Quench minimum packet gap to 100 microseconds, enter the following:

```
-> icmp type 4 code 0 min-pkt-gap 100
```

Likewise, to set the Timestamp Reply minimum packet gap to 100 microseconds, enter the following:

```
-> icmp timestamp reply min-pkt-gap 100
```

ICMP Control Table

The ICMP Control Table displays the ICMP control messages, whether they are enabled or disabled, and the minimum packet gap times. Use the **show arp send-gratuitous-arp** command to display the table.

ICMP Statistics Table

The ICMP Statistics Table displays the ICMP statistics and errors. This data can be used to monitor and troubleshoot IP on the switch. Use the **show icmp statistics** command to display the table.

Using the Ping Command

The **ping** command is used to test whether an IP destination can be reached from the local switch. This command sends an ICMP echo request to a destination and then waits for a reply. To ping a destination, enter the **ping** command and enter either the IP address of the destination or the host name. The switch pings the destination by using the default frame count, packet size, interval, and time-out parameters (6 frames, 64 bytes, 1 second, and 5 seconds, respectively). For example:

```
-> ping 172.22.2.115
```

When you ping a device, the device IP address or host name is required. Optionally, you can also specify:

- **Count.** Use the **count** keyword to set the number of frames to be transmitted.
- **Size.** Use the **size** keyword to set the size, in bytes, of the data portion of the packet sent for this ping. You can specify a size or a range of sizes up to 60000.
- **Interval.** Use the **interval** keyword to set the frequency, in seconds, that the switch polls the host.
- **Time-out.** Use the **time-out** keyword to set the number of seconds the program waits for a response before timing out.
- **source-interface.** Use the **source-interface** keyword to set the IP address to be used as source IP for the ping packets.
- **data-pattern.** Use the **data-pattern** keyword to set the data pattern to be used in the data field of the ping packets.
- **dont-fragment.** Use the **dont-fragment** keyword to set the don't-fragment bit in the IP packet.
- **tos.** Use the **tos** keyword to set the type of service field in the IP header.

For example, to send a ping with a count of 2, a size of 32 bytes, an interval of 2 seconds, time-out of 10 seconds, a source-interface using mgmt, tos of 1, data-pattern of AB and dont-fragment you would enter:

```
-> ping 172.22.2.115 count 2 size 32 interval 2 timeout 10 source-interface mgmt
tos 1 data-pattern AB dont-fragment
```

Note. If you change the default values, they only apply to the current ping. The next time you use the **ping** command, the default values are used unless you enter different values again.

Tracing an IP Route

The **traceroute** command is used to find the path taken by an IP packet from the local switch to a specified destination. This command displays the individual hops to the destination as well as timing information. When using this command, enter the name of the destination as part of the command line (either the IP address or host name). Use the optional **max-hop** parameter to set a maximum hop count to the destination. If the trace reaches this maximum hop count without reaching the destination, the trace stops.

For example, to perform a traceroute to a device with an IP address of 172.22.2.115 with a maximum hop count of 10 you would enter:

```
-> traceroute 172.22.2.115 max-hop 10
```

Optionally, you can also specify:

- **min-hop.** Use the **min-hop** keyword to set the minimum number of hops for the first packet.
- **source-interface.** Use the **source-interface** keyword to set the source IP interface to be used in the traceroute packets.
- **probes.** Use the **probes** keyword to set the number of packets (retry) to be sent for each hop-count.
- **timeout.** Use the **timeout** keyword to set the time to wait for the response of each probe packet.
- **port.** Use the **port** keyword to set the destination port number to be used in the probing packets.

Transmission Control Protocol (TCP)

TCP Half-open Timeout Configuration

Use the **ip tcp half-open-timeout** command to configure the timeout periods for dropping half-open TCP connections.

Current supported values are 3, 7, 15, 31 and 63 (in seconds). The default value is 63 seconds.

```
-> ip tcp half-open-timeout 7
```

The **show ip tcp half-open-timeout** displays the timeout value configured for half-open TCP sessions.

Displaying TCP Information

Use the **show tcp statistics** command to display TCP statistics. Use the **show tcp ports** command to display TCP port information.

Displaying UDP Information

UDP is a secondary transport-layer protocol that uses IP for delivery. UDP is not connection-oriented and does not provide reliable end-to-end delivery of datagrams. Few applications can safely use UDP to send datagrams that do not require the extra overhead added by TCP. Use the **show udp statistics** command to display UDP statistics. Use the **show udp ports** command to display UDP port information.

Tunneling

Tunneling is a mechanism that can encapsulate a wide variety of protocol packet types and route them through the configured tunnels. Tunneling is used to create a virtual point-to-point link between routers at remote points in a network. This feature supports the creation, administration, and deletion of IP interfaces whose underlying virtual device is a tunnel. The OmniSwitch implementation provides support for two tunneling protocols: Generic Routing Encapsulation (GRE) and IP encapsulation within IP(IPIP).

Generic Routing Encapsulation

GRE encapsulates a packet to be carried over the GRE tunnel with a GRE header. The resulting packet is then encapsulated with an outer header by the delivery protocol and forwarded to the other end of the GRE tunnel. The destination IP address field in the outer header of the GRE packet contains the IP address of the router at the remote end of the tunnel. The router at the receiving end of the GRE tunnel extracts the original payload and routes it to the destination address specified in the IP header of the payload.

Note. A switch can support up to 127 GRE tunnel interfaces.

IP Encapsulation within IP

IPIP tunneling is a method by which an IP packet is encapsulated within another IP packet. The Source Address and Destination Address of the outer IP header identifies the endpoints of tunnel. Whereas Source Address and Destination Address of the inner IP header identifies the original sender and recipient of the packet, respectively.

Consider the following when configuring the IPIP tunnel interfaces:

- A switch can support up to 127 IPIP tunnel interfaces.
- IPIP tunnel interfaces are included in the maximum number of IP interfaces that are supported on the switch.

Tunneling Operation

The following diagram illustrates how packets are forwarded over the tunnel.

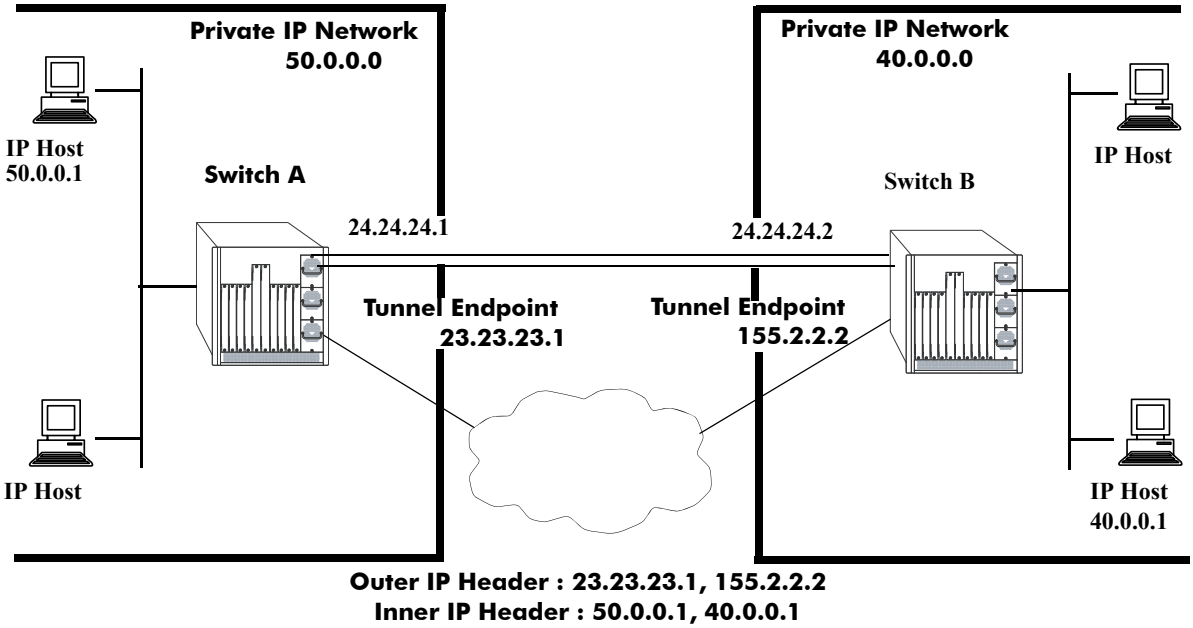


Figure 17-6 : Tunneling Operation

In the given diagram, IP packets flowing from the private IP network 50.0.0.0 to the private IP network 40.0.0.0 are encapsulated by the tunneling protocol at switch A and forwarded to switch B. Intermediate switches route the packets using addresses in the delivery protocol header. Switch B extracts the original payload and routes it to the appropriate destination in the 40.0.0.0 network.

The tunnel interface is identified as being up when all of the following are satisfied:

- Both source and destination addresses are assigned.
- The source address of the tunnel is one of the switch's IP interface addresses that is either a VLAN or Loopback0 interface.
- A route is available to reach the destination IP address. A route whose egress interface is a VLAN-based interface is available for its destination IP address. The switch supports assigning an IP address as well as routes to a tunnel interface.

This section describes how to configure a tunnel interface using GRE and IPIP, using Command Line Interface (CLI) commands.

Configuring a Tunnel Interface

To configure a GRE tunnel, use the **ip interface tunnel** command as shown:

```
-> ip interface "gre" tunnel source 23.23.23.1 destination 155.2.2.2 protocol gre
```

In this example, the GRE tunnel named “gre” is created and assigned a source IP address of 23.23.23.1 and a destination IP address of 155.2.2.2.

You can configure an IP address for the GRE tunnel interface using the **ip interface** command as shown:

```
-> ip interface "gre" address 24.24.24.1 mask 255.255.255.0
```

To configure an IPIP tunnel, use the **ip interface tunnel** command as shown:

```
-> ip interface "ipip" tunnel source 23.23.23.1 destination 155.2.2.2 protocol
ipip
```

In this example, the IPIP tunnel named “ipip” is created and assigned a source IP address of 23.23.23.1 and a destination IP address of 155.2.2.2.

You can configure an IP address for the IPIP tunnel interface using the **ip interface** command as shown:

```
-> ip interface "ipip" address 24.24.24.1 mask 255.255.255.0
```

Notes.

- An interface can be configured only as a VLAN or a Tunnel interface.
- To display information about the configured tunnels on the switch, use the **show ip interface** command.

Verifying the IP Configuration

A summary of the show commands used for verifying the IP configuration is given here:

show ip interface	Displays the usability status of interfaces configured for IP.
show ip routes	Displays the IP Forwarding table.
show ip route-pref	Displays the configured route preference of a router.
show ip router database	Displays a list of all routes (static and dynamic) that exist in the IP router database.
show ip config	Displays IP configuration parameters.
show ip protocols	Displays switch routing protocol information and status.
show ip router-id	Displays the status of TCP/UDP service ports. Includes service name and well-known port number.
show arp	Displays the ARP table.
show arp send-gratuitous-arp	This command allows the viewing of the ICMP control settings.
show ip dos config	Displays the configuration parameters of the DoS scan for the switch.
show ip dos statistics	Displays the statistics on detected port scans for the switch.
show ip dos arp-poison	Displays the number of attacks detected for a restricted address.

For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

VRF Route Leak

VRF provides isolation of routing instances from each other. The basic principle of VRF is to exclude two or more routing domains mutually by containing the exchange of routing information and forwarding packets within the same routing instance. VRF provides independent routing instances logically separating Layer3 topology of unrelated entities sharing a single physical infrastructure.

However, network devices in one VRF might need to access selected network devices in another VRF, such as in the following scenarios:

- In an enterprise, various departments can be isolated within individual VRFs but users in all the VRFs need access to the Mail Server/common enterprise portal.
- Users in other VRFs need Internet access that is available in only one VRF.
- Buildings where multiple companies sharing the same router reside within individual VRFs have to access common services like logistics, common network equipment that is a part of an independent VRF.

The VRF Route Leak feature can be used to forward routes from one VRF routing table to another VRF routing table, allowing routing from one VRF to a gateway in another VRF.

Quick Steps for Configuring VRF Route Leak

The following steps provide a quick tutorial on how to configure VRF Route Leak. Each step describes a specific operation and provides the CLI command syntax for performing that operation.

1 Create a route map to use as a filter for exporting routes using the **ip route-map action** command. For example:

```
-> ip route-map R1 action permit
```

2 Define protocol preference for export policy route map using the **ip route-map match protocol** command. This route map controls the export of routes from the VRF FDB (Forwarding Routing Database) to the GRT (Global Routing Table). A route map with no specific match clause matches all FDB routes. For example,

```
-> ip route-map R1 match protocol static
```

3 Export routes from the source VRF to the GRT using the **ip export** command. For example,

```
-> ip export route-map R1
```

4 Create a route map to use as a filter for importing routes using the **ip route-map action** command. For example:

```
-> ip route-map R2 action permit
```

5 Define protocol preference for import policy route map using the **ip route-map match protocol** command. This route map controls the import of routes from the GRT. For example:

```
-> ip route-map R2 match protocol static
```

6 Import the leaked routes from the GRT using the **ip import** command. For example,

```
-> ip import vrf V1 import route-map R2
```

7 Configure route preference for imported routes using the **ip route-pref** command with the **import** parameter. For example:

```
-> ip route-pref import 100
```

8 Redistribute imported routes to other routing protocols that are imported and added to the RDB from other VRFs using the **ip redist** command. For example:

```
-> ip redist import into ospf route-map R3 status enable
```

Configuring VRF Route Leak

This section describes how to configure VRF Route Leak using the CLI commands.

Export Routes to the GRT

Export routes from the source VRF to the Global Routing Table (GRT). Use route map to filter routes. Only those FDB (Forwarding Routing Database) routes that match the conditions of the route map are exported to GRT.

If VRF is not configured, the routes are exported from the default VRF to GRT. Only one-route map can be configured as export policy in a VRF. Route leaking between VRFs only supports IPv4 routes.

To export routes from the default VRF, enter the **ip export** command at the CLI prompt as shown:

```
-> ip export route-map R1
```

To export routes from a specific VRF, specify the VRF globally or enter into the specific VRF instance and enter **ip export** command:

```
-> vrf vrf2 ip export route-map R1

-> vrf vrf1
vrf1::-> ip export route-map R1
```

Note. To filter exported routes, create a route map and define protocol preference for the route map by using the **ip route-map** commands. A route map configured for an export policy can contain any of the following filter and set options:

- Filter options: ip-address, ip-next-hop, tag, protocol, ipv4-interface, metric, route-type, name
- Set option: tag, metric

If the tag or metric set option is *not* used in the export route map, the existing tag or metric value associated with the route is passed through unchanged. For example, a route tag is passed to the GRT unchanged unless the value is reset by a tag set clause in the export route map. For route map configuration and match extensions, see "Using Route Maps" on page 17-20.

To export all routes without a filter, use the **ip export** command with the **all-routes** parameter option.

To disable exporting of routes from the VRF to the GRT, use the **no** form of this command as shown:

```
-> no ip export R1
```

Import Routes from the GRT

Import routes from GRT to the destination VRF. Use route map to filter imported routes. Only one route map can be configured for an import policy for each export VRF.

Note. To filter imported routes, create a route map and define protocol preference for the route map by using **ip route-map** commands. A route map configured for the import policy can contain any of the following filter and set options:

- Filter options: ip-address, ip-next-hop, tag, metric
- Set option: tag, metric

For route map configuration and match extensions, "Using Route Maps" on page 17-20.

To import all routes without a filter, use the **ip import** command with the **all-routes** parameter option.

To import routes from the GRT to the destination VRF, enter the **ip import** command at the CLI prompt as shown:

```
-> ip import vrf V1 route-map R2
```

To disable importing of routes from the GRT, use the **no** form of this command as shown:

```
-> no ip import VRF V1
```

Configure Route Preference for Imported Routes

To configure the route preference for the routes that are imported and added to the RDB from other VRFs, use the **ip route-pref** command with the **import** parameter. For example,

```
-> ip route-pref import 100
```

Leaked routes are only for forwarding. If a local route is leaked, that interface is not accessible in the importing VRF. Another switch will not be able to ping the interface in the import VRF.

Redistribute Imported Routes

To enable redistribution of imported routes that are imported and added to the RDB from other VRFs into routing protocols in the routing instance, use the **ip redist** command. For example,

```
-> ip redist import into ospf route-map R3 status enable
```

Verifying VRF Route Leak Configuration

A summary of the commands used for verifying the VRF Route Leak configuration is given here:

show ip export	Displays the export route configuration details.
show ip import	Displays the import route configuration details.
show ip global-route-table	Displays the GRT for all the routes that are exported from the VRFs.

The imported routes are also displayed under the protocol field as IMPORT in the **show ip routes**, **show ip route-pref**, **show ip redist**, and **show ip router database** commands.

For more information about the output details that result from the **show** commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

18 Configuring Multiple VRF

Multiple Virtual Routing and Forwarding (VRF) provides a mechanism for segmenting Layer 3 traffic into virtual routing domains (instances) on the same switch. Each routing instance independently maintains its own routing and forwarding table, peer, and interface information.

In This Chapter

This chapter describes the Multiple VRF feature and how to configure it through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. This chapter provides an overview of Multiple VRF and includes the following information:

- "VRF Defaults" on page 18-2.
- "Quick Steps for Configuring Multiple VRF" on page 18-2.
- "Multiple VRF Overview" on page 18-5.
- "VRF Interaction With Other Features" on page 18-10.
- "Configuring VRF Instances" on page 18-14.
- "Verifying the VRF Configuration" on page 18-17.

VRF Defaults

Parameter Description	Command	Default Value/Comments
Active VRF instance	**vrf**	Default VRF instance with max profile capabilities.

Quick Steps for Configuring Multiple VRF

The initial configuration for an OmniSwitch consists of a default VRF instance. This instance is always available and is not removable. The following procedure provides a quick tutorial for creating two additional VRF instances and configuring IPv4 protocols to run in each instance:

Note. Configuring a VRF instance name is case sensitive. As a result, it is possible to accidentally create or delete instances. Use the **show vrf** command to verify the VRF instance configuration before selecting, adding, or removing instances.

1 Create VRF instance, *IpOne*, using the **vrf** command with the **create** parameter. For example:

```
-> vrf create IpOne
IpOne::->
```

In this example, the change in the command prompt from "`->`" to "`IpOne: ->`" indicates that the instance was created and is now the active VRF CLI context. Any commands entered at this point apply to this instance, unless the commands entered are not supported in multiple VRF instances.

2 Create a second VRF instance, *IpTwo*, using the **vrf** command. For example:

```
IpOne::-> vrf create IpTwo
IpTwo::->
```

In this example, *IpOne* was the active instance until *IpTwo* was created and replaced *IpOne* as the active VRF CLI context.

3 Select *IpOne* for the active VRF instance and create an IP router interface on VLAN 100 and VLAN 101 using the **ip interface** command. For example:

```
IpTwo::-> vrf IpOne
IpOne::-> ip interface intf100 address 100.1.1.1/24 vlan 100
IpOne::-> ip interface intf101 address 101.1.1.1/24 vlan 101
IpOne::->
```

4 Configure 1.1.1.1 as the primary router ID address for the *IpOne* VRF instance using the **ip router router-id** command. For example:

```
IpOne::-> ip router router-id 1.1.1.1
IpOne::->
```

5 Create an IP static route for the *IpOne* VRF instance using the **ip static-route** command. For example:

```
IpOne::-> ip static-route 192.100.1.1/24 gateway 100.1.1.10
IpOne::->
```

6 Load and enable the RIP protocol for the *IpOne* VRF instance using the **ip load rip** and **ip rip admin-state** commands. For example:

```
IpOne::-> ip load rip
IpOne::-> ip rip admin-state enable
IpOne::->
```

7 Enable RIP on IP interface "intf100" in the *IpOne* VRF instance using the **ip rip interface admin-state** command. For example:

```
IpOne::-> ip rip interface intf100 admin-state enable
IpOne::->
```

8 Select *IpTwo* for the active VRF instance and create an IP router interface on VLAN 102 using the **ip interface** command. For example:

```
IpOne::-> vrf IpTwo
IpTwo::-> ip interface intf102 address 102.1.1.1/24 vlan 102
IpTwo::->
```

9 Configure 2.2.2.2 as the primary router ID address for the *IpTwo* VRF instance using the **ip router router-id** command. For example:

```
IpTwo::-> ip router router-id 2.2.2.2
IpTwo::->
```

10 Load and enable the BGP protocol for the *IpTwo* VRF instance using the **ip load bgp** command. For example:

```
IpTwo::-> ip load bgp
IpTwo::->
```

11 Configure a BGP neighbor for the *IpTwo* VRF instance using the **ip bgp neighbor**, **ip bgp neighbor remote-as**, and **ip bgp neighbor admin-state** commands. For example:

```
IpTwo::-> ip bgp neighbor 102.1.1.10
IpTwo::-> ip bgp neighbor 102.1.1.10 remote-as 1000
IpTwo::-> ip bgp neighbor 102.1.1.10 status enable
```

12 *Optional*. To configure a VRF instance as a low profile VRF (restricted routing protocols and capabilities) use the **vrf** command with the **profile low** parameter option. For example:

```
IpTwo::-> vrf IpThree profile low
IpThree::->
```

By default, a VRF instance is created using max profile capabilities. Low profile VRFs use less switch resources, which allows more VRF instances to operate on the switch.

Note. Verify the Multiple VRF configuration using the **show vrf** command:

```
IpOne::-> show vrf
Virtual Routers     Profile Protocols
--------------------+-------+-------------------
default              default BGP PIM VRRP
IpOne                max     RIP
IpTwo                max     BGP
IpThree              low

Total Number of Virtual Routers: 4
```

To verify the configuration of a protocol within a VRF instance, use the **show** commands related to that protocol. For example, the **show ip interface** command displays the IP interfaces associated with the current CLI VRF context:

```
-> vrf IpOne
IpOne: -> show ip interface
Total 1 interfaces
        Name            IP Address     Subnet Mask   Status Forward  Device
--------------------+---------------+---------------+------+-------+--------
intfone              200.1.1.1       255.255.255.0  DOWN      NO   vlan 200
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in the above displays.

An example of what the Quick Steps configuration commands look like when entered sequentially on the switch:

```
-> vlan 100
-> vlan 101
-> vlan 102
-> vrf create IpOne
IpOne::-> vrf create IpTwo
IpTwo::-> vrf IpOne
IpOne::-> ip interface intf100 address 100.1.1.1/24 vlan 100
IpOne::-> ip interface intf101 address 101.1.1.1/24 vlan 101
IpOne::-> ip router router-id 1.1.1.1
IpOne::-> ip static-route 192.100.1.1/24 gateway 100.1.1.10
IpOne::-> ip load rip
IpOne::-> ip rip admin-state enable
IpOne::-> ip rip interface intf100 admin-state enable
IpOne::-> vrf IpTwo
IpTwo::-> ip interface intf102 address 102.1.1.1/24 vlan 102
IpTwo::-> ip router router-id 2.2.2.2
IpTwo::-> ip load bgp
IpTwo::-> ip bgp neighbor 102.1.1.10
IpTwo::-> ip bgp neighbor 102.1.1.10 remote-as 1000
IpTwo::-> ip bgp neighbor 102.1.1.10 admin-state enable
IpTwo::-> vrf IpThree profile low
```

Multiple VRF Overview

The Multiple Virtual Routing and Forwarding (VRF) feature provides the ability to configure separate routing instances on the same switch. Similar to using VLANs to segment Layer 2 traffic, VRF instances are used to segment Layer 3 traffic.

Some of the benefits of using the Multiple VRF feature include the following:

- Multiple routing instances within the same physical switch. Each VRF instance is associated with a set of IP interfaces and creates and maintains independent routing tables. Traffic between IP interfaces is only routed and forwarded to those interfaces that belong to the same VRF instance.
- Multiple instances of IP routing protocols, such as static, RIP, IPv4, BGPv4, and OSPFv2 on the same physical switch. An instance of each type of protocol operates within its own VRF instance.
- The ability to use duplicate IP addresses across VRF instances. Each VRF instance maintains its own IP address space to avoid any conflict with the service provider network or other customer networks.
- Separate IP routing domains for customer networks. VRF instances configured on the Provider Edge (PE) are used to isolate and carry customer traffic through the shared provider network.

This implementation of VRF functionality does not require a BGP/MPLS configuration in the provider network. Instead, VRF instances can route and forward IP traffic between customer sites using point-to-point Layer 3 protocols, such as IP-IP or GRE tunneling.

The illustration on page 18-6 shows an example of how the Multiple VRF feature is used to provide independent routing domains that isolate and carry customer traffic through the provider network. In this example:

- Each PE switch maintains more than one routing and forwarding table, in addition to the default VRF instance table.
- One VRF instance is configured on the PE switch for each customer network to which the PE is connected.
- Each interface on the PE that is connected to a customer edge (CE) switch is associated with the VRF instance configured for that customer.
- When an IP packet for Customer A is received on a PE 1 or PE 2 interface associated with VRF A, the VRF A instance determines how to route the packet through the provider backbone so that it reaches the intended Customer A destination.
- When an IP packet for Customer B is received on a PE 1, PE 2, or PE 3 interface associated with VRF B, the VRF B instance determines how to route the packet through the provider backbone so that it reaches the intended Customer B destination.
- When an IP packet for Customer C is received on a PE 1 or PE 3 interface associated with VRF C, the VRF C instance determines how to route the packet through the provider backbone so that it reaches the intended Customer C destination.

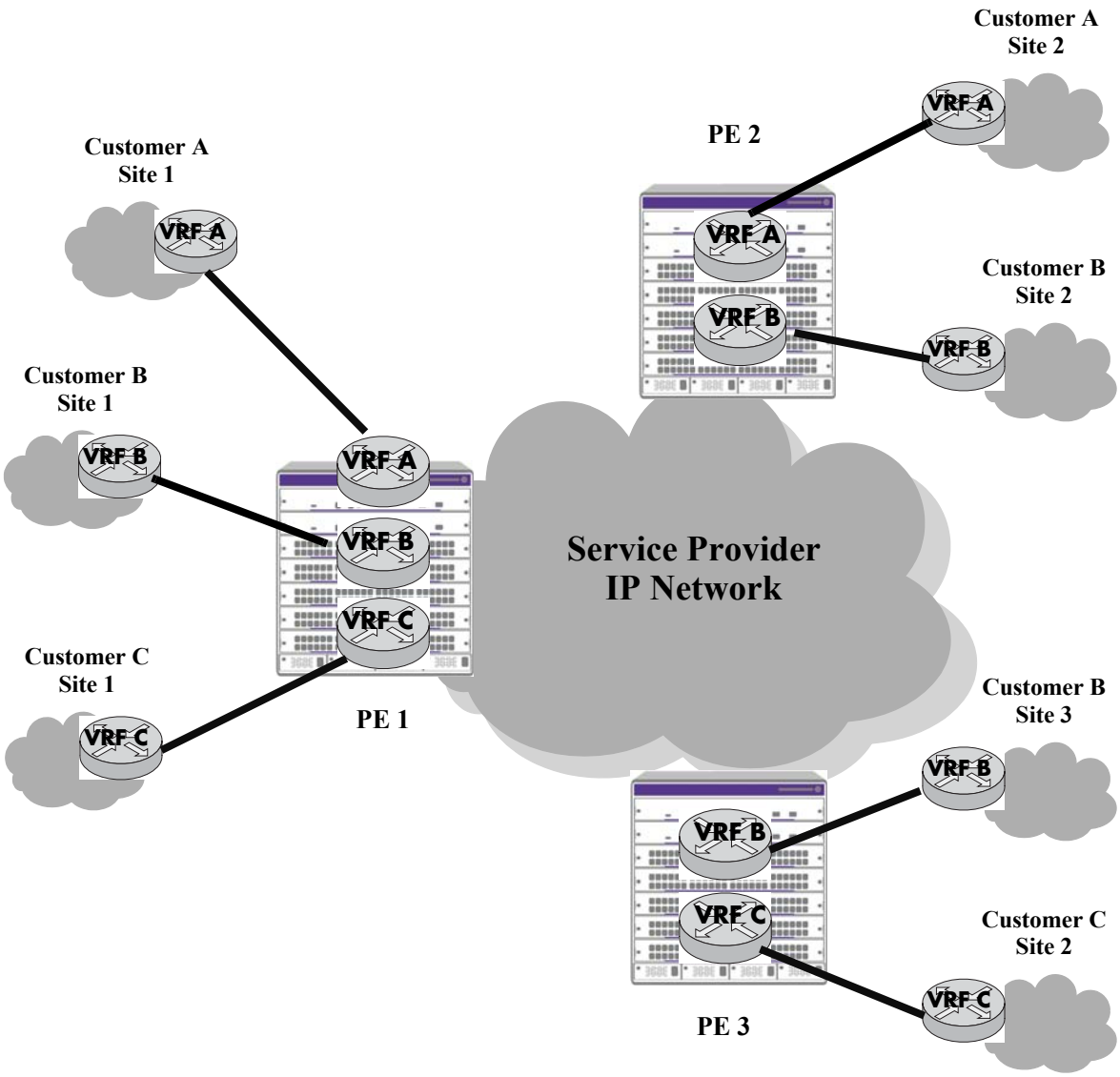


Figure 18-1 :Example Multiple VRF Configuration

VRF Profiles

The VRF feature supports two types of VRF instances: a low profile instance and a max profile instance. The type of profile assigned to a VRF instance determines the routing protocols and capabilities supported within that instance.

- Low profile VRFs only support IPv4 and VRRP, with routing capabilities restricted to static and imported routes. In addition, limiting low profiles to 9 routes and 3 IP interfaces is highly recommended.
- Max profile VRFs support full VRF routing capabilities and limits.

The type of profile applied is determined at the time the VRF instance is created. The default VRF instance uses the max profile capabilities; configuring the default VRF profile is not allowed.

Using low profile VRFs gives an administrator the ability to create VRFs with minor routing capabilities and complexity. Low profiles take up less switch resources than max profiles, which allows for creating more VRFs on the switch.

The ability to create many low profile VRFs is particularly useful in cases where all traffic only flows through a handful of individual routes to reach specific destinations; the administrator can separate many network access points into VRFs. For example: in a building there may be many tenants that need to reach several end stations and one or two WAN access points through a shared core network. Each private network needs its own address space, but does not need a routing protocol to share many routes (may only need a default route).

A combination of low and max profiles is allowed on the switch. However, the total number of VRFs allowed on the switch may differ depending on the availability of switch resources and the number of low and max profile VRFs configured.

Using the VRF Command Line Interface

The Multiple VRF feature uses a context-based command line interface (CLI). When the switch boots up, the default VRF instance is automatically created and active. Any commands subsequently entered apply to this default instance. If a different VRF instance is selected, then all subsequent commands apply to that instance.

Note. Only those commands for features that are VRF aware are accepted within the context of a VRF instance. Default VRF applications are supported only in the default VRF instance. For more information about VRF supported applications, see "VRF Interaction With Other Features" on page 18-10.

The CLI command prompt indicates which instance is the active VRF context; the instance name is added as a prefix to the command prompt. For example, if VRF instance *IpOne* is the current context, then *IpOne* appears in the CLI command prompt. For example:

```
IpOne: ->
```

When the default VRF instance is the active context, no VRF name appears in the command prompt. For example, the following prompt indicates that the default VRF instance is the current context:

```
->
```

It is also possible to enter configuration commands for other non-default instances from within the default VRF CLI context. For more information about how to do this and additional examples of using the VRF context-based CLI, see "Configuring VRF Instances" on page 18-14 and "Verifying the VRF

Configuration" on page 18-17.

Note. All VRF instances are active in terms of routing and forwarding tasks whether or not the instance is the current CLI context. Selecting a VRF instance as the CLI context simply indicates the instance to which any configuration or show commands apply.

ASCII-File-Only Syntax

When configuration commands for VRF-aware applications are configured and saved in an ASCII file (typically through the **snapshot** command) or the switch **boot.cfg** file, a prefix is added to these commands to indicate the name of the VRF instance to which the commands apply. For example:

```
! VRF
vrf create vrfOne
! IP
vrf vrfOne ip interface intf100 address 100.1.1.1/24 vlan 100
vrf vrfOne ip interface intf101 address 101.1.1.1/24 vlan 101
vrf vrfOne ip router router-id 1.1.1.1
vrf vrfOne ip static route 192.100.1.0/24 gateway 100.1.1.10
! RIP
vrf vrfOne ip load rip
vrf vrfOne ip rip status enable
vrf vrfOne ip rip interface intf100 status enable
```

In this example, *vrfOne* is added to the beginning of the IP and RIP configuration command lines. This indicates that these commands apply to the *vrfOne* instance. If a command line does not contain an instance name, then that command is for an application that applies only to the default VRF instance or the application is not VRF-aware.

Default VRF commands appear first in an ASCII or **boot.cfg** file, followed by commands for VRF-aware applications configured in non-default instances.

Management VRF

The Management VRF feature gives the user the ability to control which VRF is used for the various switch management protocols (Telnet, RADIUS, and so on.)

The following level of support is provided:

- Level 0 - The management service may only appear in the Default VRF.
- Level 1 - User may specify a single VRF that all management services can be configured in. For example, both RADIUS and LDAP can use vrf-1.
- Level 2 - Each management service or multiple management services can be configured for a different VRF. For example, RADIUS in vrf-1, LDAP in vrf-2, SNMP in vrf-3.
- Level 3 - A management service may appear in multiple VRFs. For example, SSH and Telnet in vrf-1 and vrf-2.

Level	Description	Telnet/SSH/SFTP/ SCP	Radius/SNMP/HTTP/HTTPS/ NTP/LDAP/TACACS+/Syslog
0	Default VRF Only	Yes	Yes
1	Single VRF for all services	Yes	Yes
2	Single VRF per service, each service can be on a different VRF	Yes	Yes
3	Multiple VRFs per service, any service on any VRF	Yes	No

VRF Interaction With Other Features

This section contains important information about how other OmniSwitch features interact with VRF instances. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

All OmniSwitch AOS applications fall into one of the following three categories in relation to the Multiple VRF feature:

- **VRF Aware.** Switch applications that are configurable independently and separately within one or more VRF instances. All VRF aware applications can be enabled or disabled on each VRF instance.
- **Default VRF.** Switch applications that are VRF aware but only use the default VRF instance when IP connectivity is needed; these applications are not supported across multiple VRF instances.
- **Non-VRF Aware.** Switch applications that have no association with any VRF instance, even the default instance. Note that configuration of this type of application is only allowed when the default instance is the active CLI context.

Refer to the following table to determine the VRF association for a specific switch application. Applications that do not appear in this table are non-VRF aware.

VRF-Aware Applications		Default VRF Applications	Non-VRF Aware Applications
AAA RADIUS Server BFD BGPv4 BGPv6 DVMRP FTP Server GRE Tunnels HTTP Server IPv4/ARP IP-IP Tunnels IPv6/NDP IPv6 Configured Tunnels IPv4 Multicast Switching IPv6 Multicast Switching IS-ISv4 IS-ISv6 LDAP Server NTP OSPFv2 OSPFv3 PIM-DM (IPv4)	PIM-SM (IPv4) Ping QoS VRF Policies RIPv2 RIPng Route Map Redistribution SAA IP Ping SSH Server (SSH, SFTP, SCP) SNMP (Agent) Static routes TACACS+ Server Telnet Server Traceroute UDP/DHCP Relay DHCPv6 Relay DHCP Client VRRPv2 VRRPv3 Webview	AAA DNS Client EMP access FTP Client Policy Based Routing Router Discovery Protocol sFlow SFTP SSH Client Telnet Client Trap Manager VXLAN IPv6 6to4 Tunnel	Device Profiling

The following subsections provide additional information related to Multiple VRF interaction with specific applications.

AAA RADIUS/TACACS+/LDAP Servers

- AAA RADIUS or TACACS+ or LDAP server can be configured on any VRF instance including the default VRF instance. However, all of the servers (for example, all the RADIUS servers) must reside on the same VRF instance.

- The VRF instance that the server is configured on becomes the “management” VRF instance and can perform authentication for any of the following services:

Console	**HTTP**
Telnet	**SNMP**
FTP	
SSH (ssh, sftp, and scp)	

- If the VRF instance that the servers (RADIUS / TACACS+ / LDAP) reside on is deleted or disabled, access to the servers is disabled as well.
- More than one management service can use the same VRF instance. For example, both RADIUS and and LDAP can use the same VRF instance “VrfA”.

BGPv4

- Each BGPv4 routing instance requires configuration of an Autonomous System number, router ID number, and primary IP address that is explicit to the associated VRF instance.
- BGP neighbors defined for a specific VRF instance and address family (IPv4 and IPv6) peer with neighbors accessible through interfaces associated with the same VRF instance.

IP-IP and GRE Tunnels

Tunnel endpoint addresses always exist in the default VRF instance regardless of the instance in which the tunnel interface is configured.

IPv6 Routing Protocols

IPv6 routing protocols (BGP, IS-IS, PIM, RIPng, OSPFv3, and VRRPv3) are only supported in max profile VRF instances.

Management Applications (Telnet and SSH)

- Telnet and SSH (SSH, SFTP, and SCP) sessions “to” the switch are VRF aware. Client support for these utilities is supported only in the default VRF instance.
- A maximum of four combined Telnet sessions are allowed simultaneously across all VRFs on the switch.
- A maximum of eight combined SSH sessions are allowed simultaneously across all VRFs on the switch
- More than one VRF including the default VRF can be used for Telnet / SSH sessions.

FTP

- FTP session “to” the switch is VRF aware.
- A maximum of four combined FTP sessions are allowed simultaneously across all VRFs on the switch.

NTP

Supports VRF configuration for all NTP operations (both client and server).

WebView

Supports VRF configuration for "WebView Server" and "WebView Access".

Syslog Server

Supports VRF configuration for forwarding swlog output to the syslog daemon of the switch (or host).

Quality of Service (QoS)

- The Auto-NMS feature (non-VRF aware) recognizes all of the IP interfaces configured in the default VRF instance. The first eight of these interfaces are prioritized by Auto-NMS to ensure switch manageability in the event of a DoS attack.
- Policy Based Routing, as indicated in the table above, is a default VRF application. The functionality of this feature remains the same as in releases prior to the implementation of Multiple VRF instances.

VRF Policies

- A VRF policy condition parameter is available to specify a VRF name to which the policy condition applies. This parameter can also specify the default VRF, and a **no** form of the command exists to remove a VRF condition parameter. For example:

```
-> policy condition c1 vrf engr_vrf
-> policy condition c2 vrf default
-> policy condition c1 no vrf
```

- VRF policies are configured in the default VRF, similar to how all other QoS policies are configured. If the VRF name specified does not exist, the policy is not allocated any system resources.
- Policies that do not specify a VRF name are considered global policies and are applied across all VRF instances and VLANs.
- Policies that specify the default VRF apply only to traffic in the default VRF instance.
- Policies that specify a VRF name apply only to traffic in the VRF instance associated with that name.
- The **switch** network group is supported only in VRF policies that specify the default VRF instance. If this group is specified in a global policy (no VRF specified) then the policy is applied across all VRF instances.

SNMP

- SNMPv3 is required to manage VRF instances; SNMPv1 and v2 are not supported.
- Configuring the management station to use SNMPv3 is required to receive traps from VRF-aware applications.

VLANs

Configuring an interface for a VLAN also associates that VLAN with the active VRF context. A VLAN, however, can only belong to one VRF instance at a time. As a result, all interfaces configured for a VLAN must belong to the same VRF instance. See "Assigning IP Interfaces to a VRF Instance" on page 18-16 for more information.

UDP/DHCP Relay

VRF support for UDP/DHCP Relay allows for the configuration and management of relay agents and servers within the context of a VRF instance.

The following guidelines apply when configuring UDP/DHCP Relay within the context of VRF instances:

- A separate DHCP server is required for each VRF instance to which DHCP packets are relayed to and from the server. The server should reside in the same VRF as the originating requests. For example, the following command configures the DHCP server address for the *vrfOne* instance:

  ```
  -> vrf vrfOne
  vrfOne:> ip helper address 10.0.0.1
  ```

  The above configuration relays all DHCP packets within the *vrfOne* instance to the specified server which also resides in the *vrfOne* instance.

- A separate UDP relay setting for port/service to VLAN is required per VRF instance. For example, the following command configures the forwarding of specific UDP packets to VLAN 100 within the context of the *vrfTwo* instance:

  ```
  -> ip udp dns vlan 100
  ```

- When a VRF instance is deleted, all UDP/DHCP Relay configuration associated with that instance is also deleted. However, if the VRF instance is created again with the same name, the relay configuration previously associated with that name is *not* restored.

Configuring VRF Instances

Configuring the Multiple VRF feature consists of the following:

- Creating a VRF instance with a low profile or the default max profile.
- Assigning one or more IP interfaces to the instance.
- Configuring routing protocols to operate within a specific instance.

The initial configuration of an OmniSwitch consists of a default VRF instance, which is always active when the switch starts up and is not removable from the switch configuration. Any subsequent configuration of switch applications applies only to the default instance. To provide multiple, independent IP routing domains on the same switch, configuring additional VRF instances is required.

The VRF CLI is context-based in that commands used to configure VRF-aware applications are applied to the active VRF instance. A VRF instance becomes active when the instance is either created or selected using the **vrf** command.

A VRF instance is identified by a name, which is specified at the time the instance is configured. For example, the following command creates the *IpOne* instance:

```
-> vrf create IpOne
IpOne: ->
```

In this example, instance *IpOne* is created and made the active VRF context at the same time. The CLI command prompt indicates the active context by displaying the name of the VRF instance as part of the actual prompt. Any subsequent commands entered on this command line are applied to the *IpOne* instance.

Note. Configuring a VRF instance name is case sensitive. As a result, it is possible to accidentally create or delete instances. Use the **show vrf** command to verify the VRF instance configuration before selecting, adding, or removing instances.

Within the context of the default VRF instance, it is also possible to enter configuration commands for another instance. For example, to configure an IP interface for instance *IpOne* from within the CLI context of the default instance, prefix the **ip interface** command with **vrf** command followed by the name of the instance. For example:

```
-> vrf IpOne ip interface intf100 address 100.1.1.1/24 vlan 100
->
```

The above command creates the IP interface for VRF *IpOne* but does not change the CLI context in which the command was entered. The default VRF instance remains the active context.

Note. The default VRF instance is the only VRF CLI context within which configuration of another instance is allowed.

Configuring the VRF Profile

By default, the max profile capabilities are applied when a VRF instance is created. A max profile VRF supports dynamic routing protocols and other supported VRF limits. To create a VRF instance with low profile capabilities, use the **vrf** command with the **profile low** parameter. For example:

```
-> vrf create IpTwo profile low
IpTwo-low::->
```

Changing the profile for an existing VRF instance is not allowed. To change the profile, first delete the VRF then create it again with a different profile. For example, to change profile *IpTwo* to a max profile VRF, use the following commands:

```
-> no vrf IpTwo
-> vrf create IpTwo profile max
IpTwo-low::->
```

In this example, the **profile max** parameter option is not needed, since the max profile is applied by default. However, this parameter was used here to demonstrate the command syntax.

The total number of VRFs allowed depends on the available switch memory. At 80% memory usage, a low memory warning is displayed when a new VRF is created. When 90% usage is reached, creating a new VRF is stopped. For example:

```
-> vrf create LowProfVrf400 profile low
+++ WARNING: Memory usage over 80%, creating VRF

->vrf create LowProfVrf412 profile low
ERROR: resource allocation failure
+++ ERROR: Memory usage over 90%, VRF creation failed
```

Use the **show vrf-profiles** command to display VRF profile usage information.

Selecting a VRF Instance

Moving between VRF instances is done by selecting an existing instance to become the active VRF CLI context. The **vrf** command is also used to select an existing instance. For example, the following command selects the *IpTwo* instance:

```
IpOne: -> vrf IpTwo
IpTwo: ->
```

In the above example, selecting the *IpTwo* instance changed the VRF CLI context from *IpOne* to *IpTwo*. Any subsequent commands entered apply to the *IpTwo* instance.

If the instance name specified with the **vrf** command does not exist, an error message is displayed. For example:

```
-> vrf IpFour
ERROR: VRF IpFour does not exist.
```

To return to the default VRF instance from within the context of another instance, enter the **vrf** command with or without the optional **default** parameter. For example, both of the following commands return the CLI context to the default VRF instance:

```
IpOne: -> vrf
IpOne: -> vrf default
```

Note that the command prompt for the default VRF instance does not display the instance name.

Assigning IP Interfaces to a VRF Instance

When a VRF instance is created or an existing instance is selected, any IP interface subsequently configured is associated with that instance. For example, the following commands select the *IpOne* VRF instance and configure an IP interface for that instance:

```
-> vrf IpOne
IpOne: -> ip interface intf100 address 100.1.1.1/24 vlan 100
IpOne: ->
```

Once an IP interface is associated with a VRF instance, Layer 3 traffic on that interface is routed within the domain of the VRF instance. In other words, such traffic is only routed between other IP interfaces that are associated with the same VRF instance. Any additional routing protocol traffic configured for that same interface is also routed within the associated VRF domain.

Use the following guidelines when configuring IP interfaces for a VRF instance:

- A single IP interface as well as the VLAN associated with the interface, can only belong to one VRF instance at a time.
- Once a VLAN is associated with a specific VRF instance, configuring an interface for that VLAN within the context of any other instance, is not allowed. For example, if the first IP interface configured for VLAN 100 was associated with the VRF *IpOne* instance, then any subsequent IP interface configuration for VLAN 100 is only allowed within the context of the *IpOne* instance.
- A VRF instance can have multiple VLAN associations, even though a VLAN can only have one VRF association.

Configuring Routing Protocols for a Specific VRF Instance

There are no additional CLI commands or parameters required to associate a routing protocol configuration (for example, RIP, BGP, OSPF) with a specific VRF instance. Instead, the VRF CLI context is used to determine the association between a specific routing configuration and a VRF instance. For example, if a BGP routing instance is configured when VRF instance *IpOne* is the active CLI context, then the BGP routing instance is associated with *IpOne*. All traffic for the BGP instance is routed and forwarded on the interfaces associated with VRF *IpOne*.

For more information about the interaction of switch applications with VRF instances, see "VRF Interaction With Other Features" on page 18-10. To see examples of configuring routing protocol instances within the context of a VRF instance, refer to "Quick Steps for Configuring Multiple VRF" on page 18-2.

Removing a VRF Instance

To remove a VRF instance from the switch configuration, use the **no** form of the **vrf** command. For example:

```
-> no vrf IpTwo
```

To view a list of VRF instances configured on the switch, use the **show vrf** command. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Verifying the VRF Configuration

To display a list of VRF instances configured for the switch, use the **show vrf** command. For example:

```
-> show vrf
Virtual Routers     Profile Protocols
--------------------+-------+-------------------
default              default BGP PIM VRRP
IpOne                max     RIP
IpTwo                max     BGP
IpThree              low
```

The VRF CLI context determines which information is displayed using application-specific **show** commands. For example, if *IpOne* is the active VRF context, then only IP interfaces associated with *IpOne* are displayed.

```
-> vrf IpOne
IpOne: -> show ip interface
Total 1 interfaces
        Name            IP Address     Subnet Mask   Status Forward  Device
--------------------+---------------+---------------+------+-------+--------
Loopback             127.0.0.1       255.0.0.0           UP      NO Loopback
intfone              200.1.1.1       255.255.255.0     DOWN      NO vlan 200

IpOne: -> vrf default
-> show ip interface
Total 6 interfaces
        Name            IP Address     Subnet Mask   Status Forward  Device
--------------------+---------------+---------------+------+-------+--------
EMP                  192.168.10.1    255.255.255.0     DOWN      NO EMP
Loopback             127.0.0.1       255.0.0.0           UP      NO Loopback
vlan 130             192.168.130.161 255.255.255.0     DOWN      NO vlan 130
vlan 2               10.255.11.161   255.255.255.0       UP     YES vlan 2
vlan-2000            172.20.0.1      255.255.0.0         UP     YES vlan 2000
vlan-2100            172.21.0.1      255.255.0.0         UP     YES vlan 2100
```

Note that when the default VRF CLI context is active, the **show** commands can display specific information for another instance. This is done by first entering the **vrf** command followed by the instance name and then the **show** command. For example, the following command displays the IP interfaces configured for *IpOne* from within the context of the default VRF CLI:

```
-> vrf IpOne show ip interface
```

For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

19 Configuring IPv6

Internet Protocol version 6 (IPv6) is the next generation of Internet Protocol version 4 (IPv4). Both versions are supported along with the ability to tunnel IPv6 traffic over IPv4. Implementing IPv6 solves the limited address problem currently facing IPv4, which provides a 32-bit address space. IPv6 increases the address space available to 128 bits.

In This Chapter

This chapter describes IPv6 and how to configure it through Command Line Interface (CLI). The CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

This chapter provides an overview of IPv6 and includes information about the following procedures:

- "Configuring an IPv6 Interface" on page 19-13.
- "Configuring a Unique Local IPv6 Unicast Address" on page 19-14.
- "Assigning IPv6 Addresses" on page 19-16.
- "Configuring IPv6 Tunnel Interfaces" on page 19-18.
- "Creating an IPv6 Static Route" on page 19-19.
- "Configuring the Route Preference of a Router" on page 19-21.
- "Configuring Route Map Redistribution" on page 19-22.
- "VRF Route Leak" on page 19-28
- "Configuring Local Proxy Neighbor Discovery" on page 19-31.
- "Configuring Neighbor Cache Limit" on page 19-31.
- "Configuring Neighbor Unreachability Detection" on page 19-31.
- "Configuring Router Advertisement Filtering" on page 19-33.
- "Reply or Ignore Echo Requests" on page 19-34.
- "ICMPv6 Error Message Rate Limiting" on page 19-34.
- "Configure IPv6 EMP Interface" on page 19-35.
- "Verifying the IPv6 Configuration" on page 19-37.

IPv6 Defaults

The following table lists the defaults for IPv6 configuration through the **ipv6** command.

Description	Command	Default
Global status of IPv6 on the switch	**N/A**	Enabled
Interfaces	**ipv6 interface**	loopback
6to4 tunnels	**ipv6 interface**	tunnel_6to4
Prefixes	**ipv6 prefix**	None
Hop Limit	**ipv6 hop-limit**	64
Path MTU entry minimum lifetime	**ipv6 pmtu-lifetime**	10 minutes
Neighbor stale lifetime	**ipv6 neighbor stale-lifetime**	10 minutes
Local Unicast Global ID	**ipv6 address global-id**	None

Quick Steps for Configuring IPv6 Routing

The following tutorial assumes that VLAN 200 and VLAN 300 already exist in the switch configuration. For information about how to configure VLANs, see Chapter 4, "Configuring VLANs."

1 Configure an IPv6 interface for VLAN 200 by using the **ipv6 interface** command. For example:

```
-> ipv6 interface v6if-v200 vlan 200
```

Note that when the IPv6 interface is configured, the switch automatically generates a link-local address for the interface. This allows for communication with other interfaces and/or devices on the same link, but does not provide routing between interfaces.

2 Assign a unicast address to the *v6if-v200* interface by using the **ipv6 address** command. For example:

```
-> ipv6 address 2001:db8:4100:1::/64 eui-64 v6if-v200
```

3 Configure an IPv6 interface for VLAN 300 by using the **ipv6 interface** command. For example:

```
-> ipv6 interface v6if-v300 vlan 300
```

4 Assign a unicast address to the *v6if-v300* interface by using the **ipv6 address** command. For example:

```
-> ipv6 address 2001:db8:4100:2::/64 eui-64 v6if-v300
```

Note. *Optional*. To verify the IPv6 interface configuration, enter **show ipv6 interface** For example:

```
-> show ipv6 interface
Name                 IPv6 Address/Prefix Length              Status  Device
-----------------+-----------------------------------------+-------+-----------
v6if-v200          fe80::2d0:95ff:fe12:fab5/64               Down    VLAN 200
                   2001:db8:4100:1::2d0:95ff:fe12:fab5/64
                   2001:db8:4100:1::/64
v6if-v300          fe80::2d0:95ff:fe12:fab6/64               Down    VLAN 300
                   2001:db8:4100:2::2d0:95ff:fe12:fab6/64
                   2001:db8:4100:2::/64
loopback           ::1/128                                   Active  Loopback
                   fe80::1/64
```

Note that the link-local addresses for the two new interfaces and the loopback interface were automatically created and included in the **show ipv6 interface** display output. In addition, the subnet router anycast address that corresponds to the unicast address automatically generated for the interface.

---

5 Enable RIPng for the switch by using the **ipv6 load rip** command. For example:

```
-> ipv6 load rip
```

6 Create a RIPng interface for each of the IPv6 VLAN interfaces by using the **ipv6 rip interface** command. For example:

```
-> ipv6 rip interface v6if-v200
-> ipv6 rip interface v6if-v300
```

IPv6 routing is now configured for VLAN 200 and VLAN 300 interfaces, but it is not active until at least one port in each VLAN goes active.

IPv6 Overview

IPv6 provides the basic functionality that is offered with IPv4 but includes the following enhancements and features not available with IPv4:

- **Increased IP address size**—IPv6 uses a 128-bit address, a substantial increase over the 32-bit IPv4 address size. Providing a larger address size also significantly increases the address space available, thus eliminating the concern over running out of IP addresses. See "IPv6 Addressing" on page 19-5 for more information.
- **Autoconfiguration of addresses**—When an IPv6 interface is created or a device is connected to the switch, an IPv6 link-local address is automatically assigned for the interface and/or device. See "Autoconfiguration of IPv6 Addresses" on page 19-7 for more information.
- **Anycast addresses**—A new type of address. Packets sent to an anycast address are delivered to one member of the anycast group.
- **Simplified header format**—A simpler IPv6 header format is used to keep the processing and bandwidth cost of IPv6 packets as low as possible. As a result, the IPv6 header is only twice the size of the IPv4 header despite the significant increase in address size.
- **Improved support for header options**—Improved header option encoding allows more efficient forwarding, fewer restrictions on the length of options, and greater flexibility to introduce new options.
- **Security improvements**—Extension definitions provide support for authentication, data integrity, and confidentiality.
- **Neighbor Discovery protocol**—A protocol defined for IPv6 that detects neighboring devices on the same link and the availability of those devices. Additional information that is useful for facilitating the interaction between devices on the same link is also detected (e.g., neighboring address prefixes, address resolution, duplicate address detection, link MTU, and hop limit values, etc.).

This implementation of IPv6 also provides the following mechanisms to maintain compatibility between IPv4 and IPv6:

- Dual-stack support for both IPv4 and IPv6 on the same switch.
- Configuration of IPv6 and IPv4 interfaces on the same VLAN.
- Tunneling of IPv6 traffic over an IPv4 network infrastructure.
- Embedded IPv4 addresses in the four lower-order bytes of the IPv6 address.

The remainder of this section provides a brief overview of the new IPv6 address notation, autoconfiguration of addresses, and tunneling of IPv6 over IPv4.

IPv6 Addressing

One of the main differences between IPv6 and IPv4 is that the address size has increased from 32 bits to 128 bits. Going to a 128-bit address also increases the size of the address space to the point where running out of IPv6 addresses is not a concern.

The following types of IPv6 addresses are supported:

Link-local—A link-local address is a private unicast address that identifies an interface or device on the local network. This type of address allows communication with devices and/or neighboring nodes that are attached to the same physical link. Note that when the communication is between two nodes that are not attached to the same link, both nodes must have a configured global unicast address. Routing between link-local addresses is not available because link-local addresses are not known or advertised to the general network. Link-local addresses are unique only for a link and the same link-local address may be used on multiple interfaces.

Unicast—Standard unicast addresses, similar to IPv4.

Unique Local IPv6 Unicast—IPv6 unicast address format that is globally unique and intended for local communications, usually inside of a site. These addresses are not expected to be routable on the global Internet.

Multicast—Addresses that represent a group of devices. Traffic sent to a multicast address is delivered to all members of the multicast group.

Anycast—Traffic that is sent to this type of address is delivered to one member of the anycast group. The device that receives the traffic is usually the one that is easiest to reach as determined by the active routing protocol.

Notes:

- IPv6 does not support the use of broadcast addresses. This functionality is replaced using improved multicast addressing capabilities.
- When JITC mode is enabled, Site-Local addresses of range FEC0::/10 cannot be configured. This consists of all the addresses that begin with FEC, FED, FEE and FEF. Refer to the “AAA Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information on enabling JITC mode.

IPv6 address types are identified by the high-order bits of the address, as shown in the following table:

Address Type	Binary Prefix	IPv6 Notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-local unicast	1111111010	FE80::/10
Unique Local IPv6 unicast	11111100	FC00::/7
Global unicast	everything else	

Note that anycast addresses are unicast addresses that are not identifiable by a known prefix.

IPv6 Address Notation

IPv4 addresses are expressed using dotted decimal notation and consist of four eight-bit octets. If this same method was used for IPv6 addresses, the address would contain 16 such octets, thus making it difficult to manage. IPv6 addresses are expressed using *colon hexadecimal notation* and consist of eight 16-bit words, as shown in the following example:

1234:000F:531F:4567:0000:0000:BCD2:F34A

Note that any field may contain all zeros or all ones. In addition, it is possible to shorten IPv6 addresses by suppressing leading zeros. For example:

1234:F:531F:4567:0:0:BCD2:F34A

Another method for shortening IPv6 addresses is known as *zero compression*. When an address contains contiguous words that consist of all zeros, a double colon (::) is used to identify these words. For example, using zero compression the address 0:0:0:0:1234:531F:BCD2:F34A is expressed as follows:

::1234:531F:BCD2:F34A

Because the last four words of the above address are uncompressed values, the double colon indicates that the first four words of the address all contain zeros. Note that using the double colon is only allowed once within a single address. So if the address was1234:531F:0:0:BCD2:F34A:0:0, a double colon could *not* replace both sets of zeros. For example, the first two versions of this address shown below are valid, but the last version is not valid:

1 1234:531F::BCD2:F34A:0:0

2 1234:531F:0:0:BCD2:F34A::

3 1234:531F::BCD2:F34A:: (not valid)

With IPv6 addresses that have long strings of zeros, the benefit of zero compression is more dramatic. For example, address FF00:0:0:0:0:0:4501:32 becomes FF00::4501:32.

Note that hexadecimal notation used for IPv6 addresses resembles the notation which is used for MAC addresses. However, it is important to remember that IPv6 addresses still identify a device at the Layer 3 level and MAC addresses identify a device at the Layer 2 level.

Another supported IPv6 address notation includes embedding an IPv4 address as the four lower-order bytes of the IPv6 address. This is especially useful when dealing with a mixed IPv4/IPv6 network. For example:

0:0:0:0:0:0:212.100.13.6

IPv6 Address Prefix Notation

The Classless Inter-Domain Routing (CIDR) notation is used to express IPv6 address prefixes. This notation consists of the 128-bit IPv6 address followed by a slash (/) and a number representing the prefix length (IPv6-address/prefix-length). For example, the following IPv6 address has a prefix length of 64 bits:

FE80::2D0:95FF:FE12:FAB2/64

Autoconfiguration of IPv6 Addresses

This implementation of IPv6 supports the *stateless* autoconfiguration of link-local addresses for IPv6 VLAN and tunnel interfaces and for devices when they are connected to the switch. Stateless refers to the fact that little or no configuration is required to generate such addresses and there is no dependency on an address configuration server, such as a DHCP server, to provide the addresses.

A link-local address is a private unicast address that identifies an interface or device on the local network. This type of address allows communication with devices and/or neighboring nodes that are attached to the same physical link. Note that when the communication is between two nodes that are not attached to the same link, both nodes must have a configured global unicast address. Routing between link-local addresses is not available because link-local addresses are not known or advertised to the general network.

When an IPv6 VLAN or a tunnel interface is created or a device is connected to the switch, a link-local address is automatically generated for the interface or device. This type of address consists of the well-known IPv6 prefix FE80::/64 combined with an interface ID. The interface ID is derived from the router MAC address associated with the IPv6 interface or the source MAC address if the address is for a device. The resulting link-local address resembles the following example:

FE80::2d0:95ff:fe6b:5ccd/64

Note that when this example address was created, the MAC address was modified by complementing the second bit of the leftmost byte and by inserting the hex values 0xFF and 0xFE between the third and fourth octets of the address. These modifications were made because IPv6 requires an interface ID that is derived using Modified EUI-64 format.

Stateless autoconfiguration is not available for assigning a global unicast address to an IPv6 interface. In other words, manual configuration is required to assign a non-link-local address to an interface. See "Assigning IPv6 Addresses" on page 19-16 for more information.

Both stateless and *stateful* autoconfiguration is supported for devices, such as a workstation, when they are connected to the switch. When the stateless method is used in this instance, the device listens for router advertisements in order to obtain a subnet prefix. The unicast address for the device is then formed by combining the subnet prefix with the interface ID for that device.

Stateful autoconfiguration refers to the use of an independent server, such as a DHCP server, to obtain an IPv6 unicast address and other related information. Of course, manual configuration of an IPv6 address is always available for devices as well.

Regardless of how an IPv6 address is obtained, duplicate address detection (DAD) is performed before the address is assigned to an interface or device. If a duplicate is found, the address is not assigned. Note that DAD is *not* performed for anycast addresses, 6to4 tunnels, or VRRP virtual router addresses.

Please refer to RFCs 2462, 2464, and 3513 for more technical information about autoconfiguration and IPv6 address notation.

Globally Unique Local IPv6 Unicast Addresses

These addresses are intended to be routable within a limited area such as a site but not on the global Internet. Unique Local IPv6 Unicast Addresses are used in conjunction with BGP (IBGP) speakers as well as exterior BGP (EBGP) neighbors based on configured policies. See the BGP chapter of the Advanced Routing Guide for details.

Local IPv6 unicast addresses have the following characteristics:

- Globally unique ID (with high probability of uniqueness).
- Use the well-known prefix FC00::/7 to to allow for easy filtering at site boundaries.
- Allow sites to be combined or privately interconnected without creating any address conflicts or requiring renumbering of interfaces that use these prefixes.
- Internet Service Provider independent and can be used for communications inside of a site without having any permanent or intermittent Internet connectivity.
- If accidentally leaked outside of a site via routing or DNS, there is no conflict with any other addresses.
- In practice, applications may treat these addresses like global scoped addresses.

A 40-bit global identifier is used to make the local IPv6 address prefixes globally unique. This global ID can either be explicitly configured, or created using the pseudo-algorithm recommended in RFC 4193.

Tunneling IPv6 over IPv4

It is likely that IPv6 and IPv4 network infrastructures will coexist for some time, if not indefinitely. Tunneling provides a mechanism for transitioning an IPv4 network to IPv6 and/or maintaining interoperability between IPv4 and IPv6 networks. This implementation of IPv6 supports tunneling of IPv6 traffic over IPv4. There are two types of tunnels supported, *6to4* and *configured*.

Note. Dynamic routing protocols are not supported over 6to4 tunnels. However, it is possible to configure dynamic routing for a configured tunnel. See "Configuring IPv6 Tunnel Interfaces" on page 19-18 for more information.

6to4 Tunnels

6to4 tunneling provides a mechanism for transporting IPv6 host traffic over an IPv4 network infrastructure to other IPv6 hosts and/or domains without having to configure explicit tunnel endpoints. Instead, an IPv6 6to4 tunnel interface is created at points in the network where IPv6 packets are encapsulated (IPv4 header added) prior to transmission over the IPv4 network or decapsulated (IPv4 header stripped) for transmission to an IPv6 destination.

An IPv6 6to4 tunnel interface is identified by its assigned address, which is derived by combining a 6to4 well-known prefix (2002) with a globally unique IPv4 address and embedded as the first 48 bits of an IPv6 address. For example, 2002:d467:8a89::137/64, where d467:8a89 is the hex equivalent of the IPv4 address 212.103.138.137.

6to4 tunnel interfaces are configured on routers and identify a 6to4 site. Because 6to4 tunnels are point-to-multi-point in nature, any one 6to4 router can communicate with one or more other 6to4 routers across the IPv4 cloud. Additionally, IPv6 multicast traffic cannot be forwarded over a 6to4 tunnel. Two common scenarios for using 6to4 tunnels are described below.

6to4 Site to 6to4 Site over IPv4 Domain

In this scenario, isolated IPv6 sites have connectivity over an IPv4 network through 6to4 border routers. An IPv6 6to4 tunnel interface is configured on each border router and assigned an IPv6 address with the 6to4 well-known prefix, as described above. IPv6 hosts serviced by the 6to4 border router have at least one IPv6 router interface configured with a 6to4 address. Note that additional IPv6 interfaces or external IPv6 routing protocols are not required on the 6to4 border router.

The following diagram illustrates the basic traffic flow between IPv6 hosts communicating over an IPv4 domain:

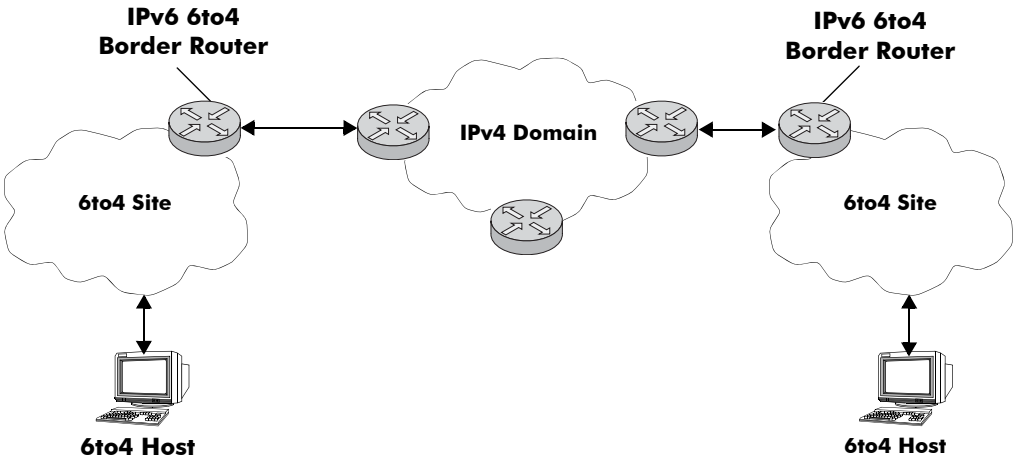


Figure 19-1 : Basic traffic flow between IPv6 hosts over IPv4 domain

In the above diagram:

1 The 6to4 hosts receive 6to4 prefix from Router Advertisement.

2 The 6to4 host sends IPv6 packets to 6to4 border router.

3 The 6to4 border router encapsulates IPv6 packets with IPv4 headers and sends to the destination 6to4 border router over the IPv4 domain.

4 The destination 6to4 border router strips IPv4 header and forwards to 6to4 destination host.

6to4 Site to IPv6 Site over IPv4/IPv6 Domains

In this scenario, 6to4 sites have connectivity to native IPv6 domains through a relay router, which is connected to both the IPv4 and IPv6 domains. The 6to4 border routers are still used by 6to4 sites for encapsulating/decapsulating host traffic and providing connectivity across the IPv4 domain. In addition, each border router has a default IPv6 route pointing to the relay router.

In essence, a relay router is a 6to4 border router on which a 6to4 tunnel interface is configured. However, a native IPv6 router interface is also required on the relay router to transmit 6to4 traffic to/from IPv6 hosts connected to an IPv6 domain. Therefore, the relay router participates in both the IPv4 and IPv6 routing domains.

The following diagram illustrates the basic traffic flow between native IPv6 hosts and 6to4 sites:

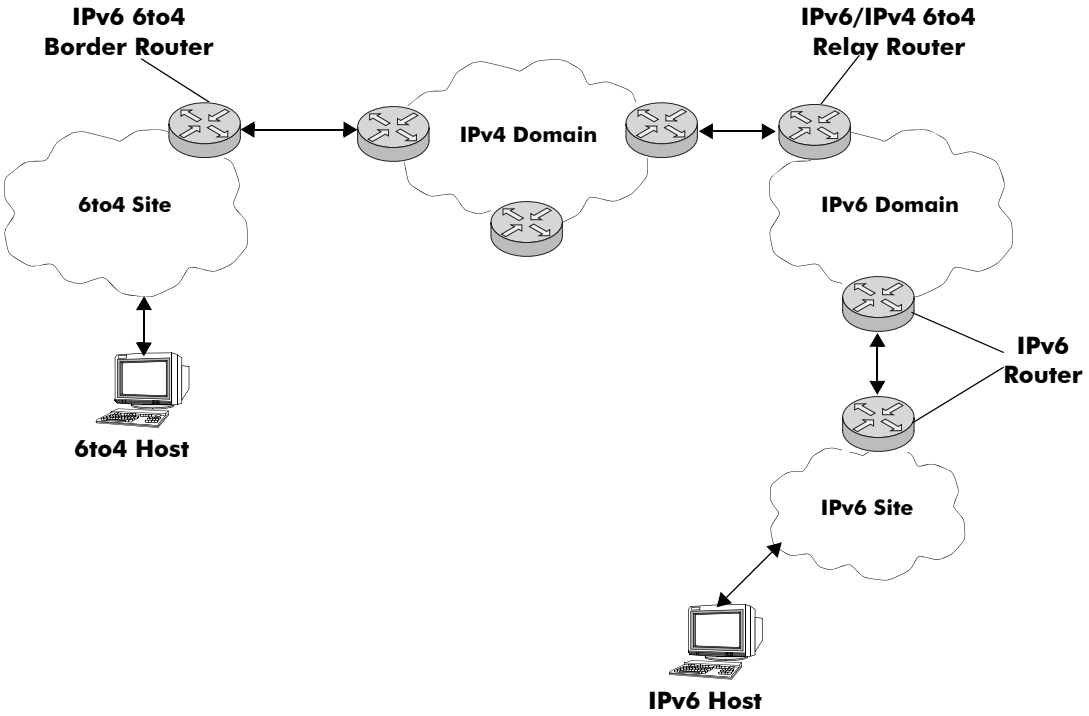


Figure 19-2 : Basic traffic flow between native IPv6 hosts and 6 to 4 sites

In the above diagram:

1 The 6to4 relay router advertises a route to 2002::/16 on its IPv6 router interface.

2 The IPv6 host traffic received by the relay router that has a next hop address that matches 2002::/16 is routed to the 6to4 tunnel interface configured on the relay router.

3 The traffic routed to the 6to4 tunnel interface is then encapsulated into IPv4 headers and sent to the destination 6to4 router over the IPv4 domain.

4 The destination 6to4 router strips the IPv4 header and forwards it to the IPv6 destination host.

For more information about configuring an IPv6 6to4 tunnel interface, see "Configuring an IPv6 Interface" on page 19-13 and "Configuring IPv6 Tunnel Interfaces" on page 19-18. For more detailed information and scenarios by using 6to4 tunnels, refer to RFC 3056.

Configured Tunnels

A configured tunnel is where the endpoint addresses are manually configured to create a point-to-point tunnel. This type of tunnel is similar to the 6to4 tunnel on which IPv6 packets are encapsulated in IPv4 headers to facilitate communication over an IPv4 network. The difference between the two types of tunnels is that configured tunnel endpoints require manual configuration, whereas 6to4 tunneling relies on an embedded IPv4 destination address to identify tunnel endpoints. Additionally, IPv6 multicast traffic can be sent over configured tunnels allows RIPng and OSPFv3 to run over a configured tunnel.

For more information about IPv6 configured tunnels, see "Configuring IPv6 Tunnel Interfaces" on page 19-18. For more detailed information about configured tunnels, refer to RFC 4213.

Local Proxy Neighbor Discovery (LPND)

Local Proxy Neighbor Discovery (LPND) is used to isolate IPv6 nodes on the same VLAN from each other. If LPND is enabled on an IPv6 VLAN interface, a client will not learn the MAC address of any other IPv6 node reached via the switch. The switch will intercept all neighbor discovery messages and replace the client MACs with the switches MAC before sending the messages to their destination. As a result, all IPv6 traffic will be routed, not switched. See "Configuring Local Proxy Neighbor Discovery" on page 19-31 for more information.

Router Advertisement (RA) Filtering

RA filtering can be used to prevent the spread of rogue RAs from unauthorized systems. If enabled on an interface, any received RAs will be dropped without being forwarded on to any other connected IPv6 clients.

One or more trusted ports or linkaggs can be specified for an interface. RAs received on those trusted ports or linkaggs will be allowed to continue on to all other IPv6 clients reached via the interface. See "Configuring Router Advertisement Filtering" on page 19-33 for more information.

Neighbor Cache Limit

The size of the neighbor cache can be limited on a system-wide basis. Once the limit is reached, no new entries will be added. The system-wide limit can be used to control the resources allocated for the IPv6 neighbor cache.

A neighbor cache limit may also be specified on a per-interface basis. Once the interface's limit is reached, no new neighbor entries are allowed. The per-interface limit can be used to prevent any particular node attached to an interface from flooding the cache, either maliciously or due to a malfunction.

By default, no limits are set (System, VRF or Interface). See "Configuring Neighbor Cache Limit" on page 19-31 for more information.

Neighbor Unreachability Detection (NUD)

IPv6 Neighbor Unreachability Detection (NUD) is performed to check the status of an unconfirmed neighbor when traffic is forwarded to it. By default, up to three neighbor solicitations are sent, with an interval of one second, to reconfirm that the neighbor is reachable.

In certain situations (e.g. high traffic loads), the default settings may not be sufficient to maintain the neighbor cache in a stable state. In such situations both the maximum number of neighbor solicitations and the interval at which they are sent may be modified. See "Configuring Neighbor Unreachability Detection" on page 19-31 for more information.

Configuring an IPv6 Interface

The **ipv6 interface** command is used to create an IPv6 interface for a VLAN or a tunnel. Note the following when configuring an IPv6 interface:

- A unique interface name is required for both a VLAN and tunnel interface.
- If creating a VLAN interface, the VLAN must already exist. See Chapter 4, "Configuring VLANs," for more information.
- If creating a tunnel interface, a tunnel ID or **6to4** is specified. Only one 6to4 tunnel is allowed per switch, so it is not necessary to specify an ID when creating this type of tunnel.
- If a tunnel ID is specified, then a configured tunnel interface is created. This type of tunnel requires additional configuration by using the **ipv6 address global-id** command. See "Configuring IPv6 Tunnel Interfaces" on page 19-18 for more information.
- Each VLAN can have one IPv6 interface. Configuring both an IPv4 and IPv6 interface on the same VLAN is allowed. Note that the VLAN interfaces of both types are not active until at least one port associated with the VLAN goes active.
- A link-local address is automatically configured for an IPv6 interface, except for 6to4 tunnels, when the interface is configured. For more information regarding how this address is formed, see "Autoconfiguration of IPv6 Addresses" on page 19-7.
- Assigning more than one IPv6 address to a single IPv6 interface is allowed.
- Assigning the same link-local address to multiple interfaces is allowed. Each global unicast prefix, subset of, or superset of a prefix can only exist on one interface. For example, if an interface for VLAN 100 is configured with an address 2001:db8:4100:1000::1/64, an interface for VLAN 200 cannot have an address 2001:db8:4100:1000::2/64.
- A subnet router anycast address is automatically created when a global unicast address is assigned to an interface.

To create an IPv6 interface for a VLAN or configured tunnel, enter **ipv6 interface** followed by an interface name, then **vlan** (or **tunnel**) followed by a VLAN ID (or tunnel ID). For example, the following two commands create an IPv6 interface for VLAN 200 and an interface for tunnel 35:

```
-> ipv6 interface v6if-v200 vlan 200
-> ipv6 interface v6if-tunnel-35 tunnel 35
```

To create an IPv6 interface for a 6to4 tunnel, use the following command:

```
-> ipv6 interface v6if-6to4 tunnel 6to4
```

Note. A 6to4 tunnel is automatically created at startup.

Use the **show ipv6 interface** command to verify the interface configuration for the switch. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring a Unique Local IPv6 Unicast Address

The **ipv6 address global-id** command is used to create a new value for the global ID. A 5-byte global ID value can be manually specified or automatically generated:

```
-> ipv6 address global-id generate
-> ipv6 address global-id 32:57a3:8fed
```

Note. If the global-id has not previously been specified with the commands above it will automatically be generated when the first IPv6 local-unicast address command is issued.

Once the global ID is generated the **ipv6 address local-unicast** command can be used to generate a unique local address using the configured global-id.

Modifying an IPv6 Interface

The **ipv6 interface** command is also used to modify existing IPv6 interface parameter values. It is not necessary to first remove the interface and then create it again with the new values. The changes specified will overwrite existing parameter values. For example, the following command changes the router advertisement (RA) reachable time and the RA retransmit timer values for interface *v6if-v200:*

```
-> ipv6 interface v6if-v200 ra-reachable-time 60000 ra-retrans-time 2000
```

When an existing interface name is specified with the **ipv6 interface** command, the command modifies specified parameters for that interface. If an unknown interface name is entered along with an existing VLAN or tunnel parameter, a new interface is created with the name specified.

Removing an IPv6 Interface

To remove an IPv6 interface from the switch configuration, use the **no** form of the **ipv6 interface** command. Note that it is only necessary to specify the name of the interface, as shown in the following example:

```
-> no ipv6 interface v6if-v200
```

Configuring an IPv6 Routed Port

In a single step, **ipv6 interface rtr-port** command creates a specified VLAN, configures an IPv6 interface for the VLAN, and assigns a port or link aggregate (tagged or untagged) to the VLAN. For example:

- To create a VLAN interface and assign port 1/1/1 as a tagged port to that VLAN, use the following command:

```
-> ipv6 interface test vlan 30 rtr-port port 1/1/1 tagged
```

- To create a VLAN interface and assign port 1/1/2 as an untagged port to that VLAN, use the following command:

```
-> ipv6 interface test1 vlan 40 rtr-port port 1/1/2 untagged
```

To configure an IPv6 routed port interface with a link aggregate, create a link aggregate first and then the VLAN interface and assign the link aggregate as tagged or untagged to that VLAN. For example:

- To create a VLAN interface and assign link aggregate 6 as tagged to that VLAN use the following command:

```
-> ipv6 interface test vlan 30 rtr-port linkagg 6 tagged
```

- To create a VLAN interface and assign link aggregate 7 as untagged to that VLAN use the following command:

```
-> ipv6 interface test1 vlan 40 rtr-port linkagg 7 untagged
```

Consider the following guidelines when creating an IPv6 routed port interface:

- Configuring an IPv4 and IPv6 routed-port interface for the same VLAN ID is supported if the following conditions are met:
  - The VLAN ID, port, and the tagged/untagged port status for both interfaces is the same (for example, IPv4 and IPv6 routed interfaces are both bound to VLAN 850 with port 1/1/2 tagged).
  - Both interfaces are configured in the same VRF instance.
- Make sure the specified VLAN ID does not already exist in the switch configuration or is only used as a routed-port VLAN for an IPv4 interface. This VLAN will serve as a routing-only VLAN with a single port or link aggregate (Layer 2 functionality is not supported).
- Make sure the specified port or link aggregate is not already assigned to a VLAN that is not a routed-port VLAN. However, the port or link aggregate can be assigned to other routed-port VLANs.
- Attempting to add more ports or link aggregates to the routed-port VLAN or attempting to delete the VLAN is not allowed. The VLAN can only be removed by deleting the associated IPv6 and, if configured, the associated IPv4 interface.
- The same VLAN cannot be used for both a routed-port interface and a non-routed port interface.
- Once configured, an IPv6 routed port interface is operationally equivalent to an IPv6 VLAN interface. Routing protocols and other switch features that use IPv6 are configured and operate on an IPv6 routed port interface in the same manner as on a regular IPv6 interface.
- If the IPv6 interface is modified such that it is no longer bound to router-port VLAN, the corresponding VLAN is deleted.

Assigning IPv6 Addresses

When an IPv6 interface is created for a VLAN or a configured tunnel, an IPv6 link-local address is automatically created for that interface. This is also true when a device, such as a workstation, is connected to the switch.

Link-local addresses, although private and non-routable, enable interfaces and workstations to communicate with other interfaces and workstations that are connected to the same link. This simplifies getting devices up and running on the local network. If this level of communication is sufficient, assigning additional addresses is not required.

If it is necessary to identify an interface or device to the entire network, or as a member of a particular group, or enable an interface to perform routing functions, then configuring additional addresses (for example, global unicast) is required.

Use the **ipv6 address** command to manually assign addresses to an existing interface (VLAN or tunnel) or device. For example, the following command assigns a global unicast address to the VLAN interface *v6if-v200*:

```
-> ipv6 address 2001:db8:4100:1000::20/64 v6if-v200
```

In the above example, 2001:db8:4100:1000:: is specified as the subnet prefix and 20 is the interface identifier. Note that the IPv6 address is expressed using CIDR notation to specify the prefix length. In the above example, **/64** indicates a subnet prefix length of 64 bits.

To use the MAC address of an interface or device as the interface ID, specify the **eui-64** option with this command. For example:

```
-> ipv6 address 2001:db8:4100:1000::/64 eui-64 v6if-v200
```

The above command example creates address 2001:db8:4100:1000::2d0:95ff:fe12:fab2/64 for interface *v6if-v200*.

Note the following when configuring IPv6 addresses:

- It is possible to assign more than one address to a single interface.
- Any field of an address may contain all zeros or all ones. The exception to this is the interface identifier portion of the address, which cannot be all zeros. If the **eui-64** option is specified with the **ipv6 address** command, this is not an issue.
- The EUI-64 interface identifier takes up the last 64 bits of the 128-bit IPv6 address. If the subnet prefix combined with the EUI-64 interface ID is longer than 128 bits, an error occurs and the address is not created.

A subnet router anycast address is automatically created when a global unicast address is assigned to an interface. The anycast address is derived from the global address by adding an interface ID of all zeros to the prefix of the global address. For example, the global address 2001:db8:4100:1000::20/64 generates the anycast address 2001:db8:4100:1000::/64.

- Devices, such as a PC, are eligible for stateless autoconfiguration of unicast addresses in addition to the link-local address. If this type of configuration is in use on the network, manual configuration of addresses on the PC is not required.
- IPv6 VLAN or tunnel interfaces are only eligible for stateless autoconfiguration of their link-local addresses. Manual configuration of addresses is required for all additional addresses.

See "IPv6 Addressing" on page 19-5 for an overview of IPv6 address notation. Refer to RFC 4291 for more technical address information.

Removing an IPv6 Address

To remove an IPv6 address from an interface, use the **no** form of the **ipv6 address** command as shown:

```
-> no ipv6 address 2001:db8:4100:1000::20 v6if-v200
```

Note that the subnet router anycast address is automatically deleted when the last unicast address of the same subnet is removed from the interface.

Configuring IPv6 Tunnel Interfaces

There are two types of tunnels supported, 6to4 and configured. Both types facilitate the interaction of IPv6 networks with IPv4 networks by providing a mechanism for carrying IPv6 traffic over an IPv4 network infrastructure. This is an important function since it is more than likely that both protocols will need to coexist within the same network for some time.

A 6to4 tunnel is configured by creating an IPv6 6to4 tunnel interface on a router. This interface is then assigned an IPv6 address with an embedded well-known 6to4 prefix (e.g., 2002) combined with an IPv4 local address. This is all done using the **ipv6 interface** and **ipv6 address** commands. Since a 6to4 interface named "tunnel_6to4" is automatically created, enter the following commands to create a 6to4 tunnel interface:

```
-> ipv6 address 2002:c633:6489::254/16 tunnel_6to4
-> ipv6 interface tunnel_6to4 admin-state enable
```

In the above example, 2002 is the well-known prefix that identifies a 6to4 tunnel. The C633:6489 part of the address that follows 2002 is the hex equivalent of the IPv4 address 198.51.100.137. Note that an IPv4 interface configured with the embedded IPv4 address is required on the switch. In addition, do not configure a private (e.g., 192.168.10.1), broadcast, or unspecified address as the embedded IPv4 address.

One of the main benefits of 6to4 tunneling is that no other configuration is required to identify tunnel endpoints. The router that the 6to4 tunnel interface is configured on will encapsulate IPv6 packets in IPv4 headers and send them to the IPv4 destination address where they will be processed. This is particularly useful in situations where the IPv6 host is isolated.

The second type of tunnel supported is referred to as a configured tunnel. With this type of tunnel it is necessary to specify an IPv4 address for the source and destination tunnel endpoints. Note that if bidirectional communication is desired, then it is also necessary to create the tunnel interface at each end of the tunnel.

Creating an IPv6 configured tunnel involves the following general steps:

- Create an IPv6 tunnel interface using the **ipv6 interface** command.
- Associate an IPv4 source and destination address with the tunnel interface by using the **ipv6 interface** command. These addresses identify the tunnel endpoints.
- Associate an IPv6 address with the tunnel interface by using the **ipv6 address** command.
- Configure a tunnel interface and associated addresses at the other end of tunnel.

The following example commands create the *v6if-tunnel-137* configured tunnel:

```
-> ipv6 interface v6if-tunnel-137 tunnel 1
-> ipv6 interface v6if-tunnel-137 tunnel source 198.51.100.137 destination
192.0.2.195
-> ipv6 address 2100:db8:4132:4000::/64 eui-64 v6if-tunnel-137
-> ipv6 interface v6if-tunnel-137 admin-state enable
```

Note that dynamic routing protocols are not supported over 6to4 tunnels, but are allowed over configured tunnels. To use this protocol on a configured tunnel, a dynamic routing protocol interface is created for the tunnel interface. For example, the following command creates a RIPng interface for tunnel v6if-tunnel-137:

```
-> ipv6 rip interface v6if-tunnel-137
```

Creating an IPv6 Static Route

Static routes are user-defined and by default, carry a higher priority than dynamic routes. That is, if two routes have the same metric value, the static route has the higher priority. Static routes allow you to define, or customize, an explicit path to an IPv6 network segment, which is then added to the IPv6 Forwarding table. Static routes can be created between VLANs to enable devices on these VLANs to communicate.

Use the **ipv6 static-route** command to create a static route. You must specify the destination IPv6 address of the route as well as the IPv6 address of the first hop (gateway) used to reach the destination. For example, to create a static route to IPv6 address 212:95:5::/64 through gateway `fe80::2d0:95ff:fe6a:f458` on interface v6if-137, you would enter:

```
-> ipv6 static-route 2001:db8:212:95::/64 gateway fe80::2d0:95ff:fe6a:f458 v6if-137
```

Note that in the example above the IPv6 interface name for the gateway was included. This parameter is required only when a link local address is specified as the gateway.

It is possible to configure an IPv6 blackhole route. A blackhole route is used to forward unwanted traffic to a black-hole. Static routes may be created for undesirable destinations by pointing them to a NULL interface instead of valid gateway address. Any traffic that has a destination matching this undesirable destination shall be dropped automatically in hardware without going to the CPU.

Use the **null** option in the **ipv6 static-route** command to create an IPv6 blackhole route:

```
-> ipv6 static-route 212:95:5::/64 gateway null
```

Alternatively, the gateway address '::' can be used to create an IPv6 blackhole route.

```
-> ipv6 static-route 22::/64 gateway ::
```

When you create a static route, the default metric value of 1 is used. However, you can change the priority of the route by increasing its metric value. The lower the metric value, the higher the priority. This metric is added to the metric cost of the route. The metric range is 1 to 15. For example:

```
-> ipv6 static-route 2001:db8:212:95::/64 gateway fe80::2d0:95ff:fe6a:f458 v6if-137
metric 3
```

Static routes do not age out of the IPv6 Forwarding table; you must delete them from the table. Use the **no ipv6 static-route** command to delete a static route. You must specify the destination IPv6 address of the route as well as the IPv6 address of the first hop (gateway). For example, to delete the static you would enter:

```
-> no ip static-route 2001:db8:212:95::/64 gateway fe80::2d0:95ff:fe6a:f458 v6if-137
```

The IPv6 Forwarding table includes routes learned through one of the routing protocols (RIP, OSPF, BGP) as well as any static routes that are configured. Use the **show ipv6 routes** command to display the IPv6 Forwarding table.

Note. A static route is not active unless the gateway it is using is active.

To create an IPv6 static route with gateway pointing to an EMP interface, specify the keyword **emp** for the interface name field instead of specifying the exact interface name of the EMP interface (for example, EMP-CMMA-CHAS1).

```
-> ipv6 static-route 212:95:5::/64 gateway 2001::205 emp

or
```

```
-> ipv6 static-route 212:95:5::/64 gateway 2001::205 EMP-CMMA-CHAS1
```

Note.

- If an IPv6 address is configured on the EMP port on a VC or chassis setup, ensure to configure the IPv6 address on all the VC elements or CMM.
- Static route with default gateway towards EMP interface is not allowed.

```
-> ipv6 static-route ::/0 gateway 2001::205 EMP-CMMA-CHAS1
ERROR: Default routes with gateway on EMP port not allowed
```

See "Configure IPv6 EMP Interface" on page 19-35 for more information on configuring IPv6 EMP interface.

Configuring the Route Preference of a Router

By default, the route preference of a router is in this order: local, static, OSPFv3, RIPng, EBGP, and IBGP (highest to lowest).

Use the **ipv6 route-pref** command to change the route preference value of a router. For example, to configure the route preference of an OSPF route, you would enter:

```
-> ipv6 route-pref ospf 15
```

To display the current route preference configuration, use the **show ipv6 route-pref** command:

```
-> show ipv6 route-pref
  Protocol    Route Preference Value
------------+------------------------
  Local                 1
  Static                2
  OSPF                110
  ISISL1              115
  ISISL2              118
  RIP                 120
  EBGP                190
  IBGP                200
  Import              210
```

Configuring Route Map Redistribution

It is possible to learn and advertise IPv6 routes between different protocols. Such a process is referred to as route redistribution and is configured using the **ipv6 redist** command.

Redistribution uses route maps to control how external routes are learned and distributed. A route map consists of one or more user-defined statements that can determine which routes are allowed or denied access to the receiving network. In addition a route map may also contain statements that modify route parameters before they are redistributed.

When a route map is created, it is given a name to identify the group of statements that it represents. This name is required by the **ipv6 redist** command. Therefore, configuring route redistribution involves the following steps:

1 Create a route map, as described in "Using Route Maps" on page 19-22.

2 Configure redistribution to apply a route map, as described in "Configuring Route Map Redistribution" on page 19-26.

Using Route Maps

A route map specifies the criteria that are used to control redistribution of routes between protocols. Such criteria is defined by configuring route map statements. There are three different types of statements:

- **Action.** An action statement configures the route map name, sequence number, and whether or not redistribution is permitted or denied based on route map criteria.
- **Match.** A match statement specifies criteria that a route must match. When a match occurs, then the action statement is applied to the route.
- **Set.** A set statement is used to modify route information before the route is redistributed into the receiving protocol. This statement is only applied if all the criteria of the route map is met and the action permits redistribution.

The **ip route-map** command is used to configure route map statements and provides the following **action**, **match**, and **set** parameters:

ip route-map action ...	ip route-map match ...	ip route-map set ...
permit **deny**	**ip-address** **ip-nexthop** **ipv6-address** **ipv6-nexthop** **tag** **ipv4-interface** **ipv6-interface** **metric** **route-type** **protocol** **name**	**metric** **metric-type** **tag** **community** **local-preference** **level** **ip-nexthop** **ipv6-nexthop**

Refer to the "IP Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the **ip route-map** command parameters and usage guidelines.

Once a route map is created, it is then applied using the **ipv6 redist** command. See "Configuring Route Map Redistribution" on page 19-26 for more information.

Creating a Route Map

When a route map is created, it is given a name (up to 20 characters), a sequence number, and an action (permit or deny). Specifying a sequence number is optional. If a value is not configured, then the number 50 is used by default.

To create a route map, use the **ip route-map** command with the **action** parameter. For example,

```
-> ip route-map ospf-to-rip sequence-number 10 action permit
```

The above command creates the ospf-to-rip route map, assigns a **sequence number** of 10 to the route map, and specifies a **permit** action.

To optionally filter routes before redistribution, use the **ip route-map** command with a **match** parameter to configure match criteria for incoming routes. For example,

```
-> ip route-map ospf-to-rip sequence-number 10 match tag 8
```

The above command configures a match statement for the ospf-to-rip route map to filter routes based on their tag value. When this route map is applied, only OSPF routes with a tag value of eight are redistributed into the RIP network. All other routes with a different tag value are dropped.

Note. Configuring match statements is not required. However, if a route map does not contain any match statements and the route map is applied using the **ipv6 redist** command, the router redistributes *all* routes into the network of the receiving protocol.

To modify route information before it is redistributed, use the **ip route-map** command with a **set** parameter. For example,

```
-> ip route-map ospf-to-rip sequence-number 10 set tag 5
```

The above command configures a set statement for the ospf-to-rip route map that changes the route tag value to five. Because this statement is part of the ospf-to-rip route map, it is only applied to routes that have an existing tag value equal to eight.

The following is a summary of the commands used in the above examples:

```
-> ip route-map ospf-to-rip sequence-number 10 action permit
-> ip route-map ospf-to-rip sequence-number 10 match tag 8
-> ip route-map ospf-to-rip sequence-number 10 set tag 5
```

To verify a route map configuration, use the **show ip route-map** command:

```
-> show ip route-map
Route Maps: configured: 1 max: 200
Route Map: ospf-to-rip Sequence Number: 10 Action permit
  match tag 8
  set tag 5
```

Deleting a Route Map

Use the **no** form of the **ip route-map** command to delete an entire route map, a route map sequence, or a specific statement within a sequence.

To delete an entire route map, enter **no ip route-map** followed by the route map name. For example, the following command deletes the entire route map named redistipv4:

```
-> no ip route-map redistipv4
```

To delete a specific sequence number within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the actual number. For example, the following command deletes sequence 10 from the redistipv4 route map:

```
-> no ip route-map redistipv4 sequence-number 10
```

Note that in the above example, the redistripv4 route map is not deleted. Only those statements associated with sequence 10 are removed from the route map.

To delete a specific statement within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the sequence number for the statement, then either **match** or **set** and the match or set parameter and value. For example, the following command deletes only the match tag 8 statement from route map redistipv4 sequence 10:

```
-> no ip route-map redistipv4 sequence-number 10 match tag 8
```

Configuring Route Map Sequences

A route map may consist of one or more sequences of statements. The sequence number determines which statements belong to which sequence and the order in which sequences for the same route map are processed.

To add match and set statements to an existing route map sequence, specify the same route map name and sequence number for each statement. For example, the following series of commands creates route map rm_1 and configures match and set statements for the rm_1 sequence 10:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 set metric 1
```

To configure a new sequence of statements for an existing route map, specify the same route map name but use a different sequence number. For example, the following command creates a new sequence 20 for the rm_1 route map:

```
-> ip route-map rm_1 sequence-number 20 action permit
-> ip route-map rm_1 sequence-number 20 match ipv4-interface to-finance
-> ip route-map rm_1 sequence-number 20 set metric 5
```

The resulting route map appears as follows:

```
-> show ip route-map rm_1
Route Map: rm_1 Sequence Number: 10 Action permit
  match tag 8
  set metric 1
Route Map: rm_1 Sequence Number: 20 Action permit
  match ip4 interface to-finance
  set metric 5
```

Sequence 10 and sequence 20 are both linked to route map rm_1 and are processed in ascending order according to their sequence number value. Note that there is an implied logical OR between sequences. As

a result, if there is no match for the tag value in sequence 10, then the match interface statement in sequence 20 is processed. However, if a route matches the tag 8 value, then sequence 20 is not used. The set statement for whichever sequence was matched is applied.

A route map sequence may contain multiple match statements. If these statements are of the same kind (e.g., match tag 5, match tag 8, etc.) then a logical OR is implied between each like statement. If the match statements specify different types of matches (e.g. match tag 5, match ip4 interface to-finance, etc.), then a logical AND is implied between each statement. For example, the following route map sequence will redistribute a route if its tag is either 8 or 5:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
```

The following route map sequence will redistribute a route if the route has a tag of 8 or 5 *and* the route was learned on the IPv6 interface to-finance:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 match ipv6-interface to-finance
```

Configuring Access Lists

An IP access list provides a convenient way to add multiple IPv4 or IPv6 addresses to a route map. Using an access list avoids having to enter a separate route map statement for each individual IP address. Instead, a single statement is used that specifies the access list name. The route map is then applied to all the addresses contained within the access list.

Configuring an IP access list involves two steps: creating the access list and adding IP addresses to the list. To create an IP access list, use the **ip access-list** command (IPv4) or the **ipv6 access-list** command (IPv6) and specify a name to associate with the list. For example,

```
-> ip access-list ipaddr
-> ipv6 access-list ip6addr
```

To add addresses to an access list, use the **ip access-list address** (IPv4) or the **ipv6 access-list address** (IPv6) command. For example, the following commands add addresses to an existing access list:

```
-> ip access-list ipaddr address 10.0.0.0/8
-> ipv6 access-list ip6addr address 2001::/64
```

Use the same access list name each time the above commands are used to add additional addresses to the same access list. In addition, both commands provide the ability to configure if an address and/or its matching subnet routes are permitted (the default) or denied redistribution. For example:

```
-> ip access-list ipaddr address 16.24.2.1/16 action deny redist-control all-
subnets
-> ipv6 access-list ip6addr address 2001::1/64 action permit redist-control no-
subnets
```

For more information about configuring access list commands, see the “IP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Route Map Redistribution

The **ipv6 redist** command is used to configure the redistribution of routes from a source protocol into the destination protocol. This command is used on the IPv6 router that will perform the redistribution.

Note. A router automatically becomes an Autonomous System Border Router (ASBR) when redistribution is configured on the router.

A source protocol is a protocol from which the routes are learned. A destination protocol is the one into which the routes are redistributed. Make sure that both protocols are loaded and enabled before configuring redistribution.

Redistribution applies criteria specified in a route map to routes received from the source protocol. Therefore, configuring redistribution requires an existing route map. For example, the following command configures the redistribution of OSPFv3 routes into the RIPng network using the ospf-to-rip route map:

```
-> ipv6 redist ospf into rip route-map ospf-to-rip
```

OSPFv3 routes received by the router interface are processed based on the contents of the ospf-to-rip route map. Routes that match criteria specified in this route map are either allowed or denied redistribution into the RIPng network. The route map may also specify the modification of route information before the route is redistributed. See "Using Route Maps" on page 19-22 for more information.

To remove a route map redistribution configuration, use the **no** form of the **ipv6 redist** command. For example:

```
-> no ipv6 redist ospf into rip route-map ospf-to-rip
```

Use the **show ipv6 redist** command to verify the redistribution configuration:

```
-> show ipv6 redist

Source       Destination
Protocol     Protocol     Status    Route Map
------------+------------+---------+--------------------
localIPv6    RIPng        Enabled   ipv6rm
OSPFv3       RIPng        Enabled   ospf-to-rip
```

Configuring the Administrative Status of the Route Map Redistribution

The administrative status of a route map redistribution configuration is enabled by default. To change the administrative status, use the **admin-state** parameter with the **ipv6 redist** command. For example, the following command disables the redistribution administrative status for the specified route map:

```
-> ipv6 redist ospf into rip route-map ospf-to-rip admin-state disable
```

The following command example enables the administrative status:

```
-> ipv6 redist ospf into rip route-map ospf-to-rip admin-state enable
```

Route Map Redistribution Example

The following example configures the redistribution of OSPFv3 routes into a RIPng network using a route map (ospf-to-rip) to filter specific routes:

```
-> ip route-map ospf-to-rip sequence-number 10 action deny
-> ip route-map ospf-to-rip sequence-number 10 match tag 5
-> ip route-map ospf-to-rip sequence-number 10 match route-type external type2
-> ip route-map ospf-to-rip sequence-number 20 action permit
-> ip route-map ospf-to-rip sequence-number 20 match ipv6-interface intf_ospf
-> ip route-map ospf-to-rip sequence-number 20 set metric 255

-> ip route-map ospf-to-rip sequence-number 30 action permit
-> ip route-map ospf-to-rip sequence-number 30 set tag 8

-> ip redist ospf into rip route-map ospf-to-rip
```

The resulting ospf-to-rip route map redistribution configuration does the following:

- Denies the redistribution of Type 2 external OSPFv3 routes with a tag set to five.
- Redistributes into RIPng all routes learned on the intf_ospf interface and sets the metric for such routes to 255.
- Redistributes into RIPng all other routes (those not processed by sequence 10 or 20) and sets the tag for such routes to eight.

VRF Route Leak

VRF provides isolation of routing instances from each other. The basic principle of VRF is to exclude two or more routing domains mutually by containing the exchange of routing information and forwarding packets within the same routing instance. VRF provides independent routing instances logically separating Layer3 topology of unrelated entities sharing a single physical infrastructure.

However, network devices in one VRF might need to access selected network devices in another VRF, such as in the following scenarios:

- In an enterprise, various departments can be isolated within individual VRFs but users in all the VRFs need access to the Mail Server/common enterprise portal.
- Users in other VRFs need Internet access that is available in only one VRF.
- Buildings where multiple companies sharing the same router reside within individual VRFs have to access common services like logistics, common network equipment that is a part of an independent VRF.

The VRF Route Leak feature can be used to forward routes from one VRF routing table to another VRF routing table, allowing routing from one VRF to a gateway in another VRF.

Quick Steps for Configuring VRF Route Leak

The following steps provide a quick tutorial on how to configure VRF Route Leak. Each step describes a specific operation and provides the CLI command syntax for performing that operation.

1 Create a route map to use as a filter for exporting routes using the **ip route-map action** command. For example:

```
-> ip route-map R1 action permit
```

2 Define protocol preference for export policy route map using the **ip route-map match protocol** command. This route map controls the export of routes from the VRF FDB (Forwarding Routing Database) to the GRT (Global Routing Table). A route map with no specific match clause matches all FDB routes. For example:

```
-> ip route-map R1 match protocol static
```

3 Export routes from the source VRF to the GRT using the **ipv6 export** command. For example:

```
-> ipv6 export route-map R1
```

4 Create a route map to use as a filter for importing routes using the **ip route-map action** command. For example:

```
-> ip route-map R2 action permit
```

5 Define protocol preference for import policy route map using the **ip route-map match protocol** command. This route map controls the import of routes from the GRT. For example:

```
-> ip route-map R2 match protocol static
```

6 Import the leaked routes from the GRT using the **ipv6 import** command. For example:

```
-> ipv6 import vrf V1 route-map R2
```

7 Configure route preference for imported routes using the **ipv6 route-pref** command with the **import** parameter. For example:

```
-> ipv6 route-pref import 100
```

8 Redistribute imported routes to other routing protocols that are imported and added to the RDB from other VRFs using the **ipv6 redist** command. For example,

```
-> ipv6 redist import into ospf route-map R3 status enable
```

Configuring VRF Route Leak

This section describes how to configure VRF Route Leak using the CLI commands. Consider the following when configuring IPv6 route leaking between VRFs:

- To leak IPv6 routes between VRF instances, IPv6 must be available in both instances. It is important to note that IPv6 route leaking is supported only in max profile VRFs.
- IPv6 route leaking supports configured tunnel routes, but 6to4 tunnel and loopback routes are not supported.
- Routes that were leaked into a VRF instance cannot be exported from that instance to another VRF.

Export Routes to the GRT

Export routes from the source VRF to the Global Routing Table (GRT). Use route map to filter routes. Only those FDB (Forwarding Routing Database) routes that match the conditions of the route map are exported to GRT.

If VRF is not configured, the routes are exported from the default VRF to GRT. Only one-route map can be configured as export policy in a VRF. Route leaking between VRFs supports IPv4 and IPv6 routes.

To export routes from the default VRF, enter the **ipv6 export** command at the CLI prompt as shown:

```
-> ipv6 export route-map R1
```

To export routes from a specific VRF, specify the VRF globally or enter into the specific VRF instance and enter the **ipv6 export** command:

```
-> vrf vrf2 ipv6 export route-map R1

-> vrf vrf1
vrf1::-> ipv6 export route-map R1
```

Note. To filter exported routes, create a route map and define protocol preference for the route map by using the **ip route-map** commands. A route map configured for an export policy can contain any of the following filter and set options:

- Filter options: ipv6-address, ipv6-next-hop, tag, protocol, ipv6-interface, metric, route-type, name
- Set option: tag, metric

If the tag or metric set option is *not* used in the export route map, the existing tag or metric value associated with the route is passed through unchanged. For example, a route tag is passed to the GRT unchanged unless the value is reset by a tag set clause in the export route map. For route map configuration and match extensions, see "Using Route Maps" on page 19-22.

To export all routes without a filter, use the **ipv6 export** command with the **all-routes** parameter option.

To disable exporting of routes from the VRF to the GRT, use the **no** form of this command as shown:

```
-> no ipv6 export R1
```

Import Routes from the GRT

Import routes from GRT to the destination VRF. Use route map to filter imported routes. Only one route map can be configured for an import policy for each export VRF.

Note. To filter imported routes, create a route map and define protocol preference for the route map by using **ip route-map** commands. A route map configured for the import policy can contain any of the following filter and set options:

- Filter options: ipv6-address (no aggregates), ipv6-next-hop, tag, metric
- Set option: tag, metric

For route map configuration and match extensions, "Using Route Maps" on page 19-22.

To import all routes without a filter, use the **ipv6 import** command with the **all-routes** parameter option.

To import routes from the GRT to the destination VRF, enter the **ipv6 import** command at the CLI prompt as shown:

```
-> ipv6 import vrf V1 route-map R2
```

To disable importing of routes from the GRT, use the **no** form of this command as shown:

```
-> no ipv6 import VRF V1
```

Configure Route Preference for Imported Routes

To configure the route preference for the routes that are imported and added to the RDB from other VRFs, use the **ipv6 route-pref** command with the **import** parameter. For example,

```
-> ipv6 route-pref import 100
```

Leaked routes are only for forwarding. If a local route is leaked, that interface is not accessible in the importing VRF. Another switch will not be able to ping the interface in the import VRF.

Redistribute Imported Routes

To enable redistribution of imported routes that are imported and added to the RDB from other VRFs into routing protocols in the routing instance, use the **ipv6 redist** command. For example,

```
-> ipv6 redist import into ospf route-map R3 admin-status enable
```

Verifying VRF Route Leak Configuration

A summary of the commands used for verifying the VRF Route Leak configuration is given here:

show ipv6 export	Displays the export route configuration details.
show ipv6 import	Displays the import route configuration details.
show ipv6 global-route-table	Displays the GRT for all the routes that are exported from the VRFs.

The imported routes are also displayed under the protocol field as IMPORT in the **show ipv6 routes**, **show ipv6 route-pref**, **show ipv6 redist**, and **show ipv6 router database** commands.

For more information about the output details that result from the **show** commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Local Proxy Neighbor Discovery

To enable Local Proxy Neighbor Discovery (LPND), use the command **ipv6 interface** with the parameter **local-proxy-nd**. For example, the following command enables the LPND on the "vlan_1" IPv6 interface:

```
-> ipv6 interface vlan_1 local-proxy-nd
```

To disable LPND, use the **no** form of the **ipv6 interface** command. For example, the following command disables the LPND on the "vlan_1" IPv6 interface:

```
-> ipv6 interface vlan_1 no local-proxy-nd
```

Configuring Neighbor Cache Limit

To set the system-wide cache limit, use the **ipv6 neighbor limit** command. For example, the following command sets the system-wide cache limit to 9000 entries:

```
-> ipv6 neighbor limit 9000
```

To set the per-interface limit, use the **ipv6 interface** commands 'neighbor limit' option. For example, the following command sets the "vlan_1" interface limit to 100 entries:

```
-> ipv6 interface vlan_1 neighbor-limit 100
```

Note. To view the information on IPV6 cache limit, use the **show ipv6 information** command.

Configuring Neighbor Unreachability Detection

To specify the maximum number of neighbor solicitations to be sent during the Neighbor Unreachability Detection (NUD) process, use the **ipv6 interface** command with the **retrans-max** parameter. For example:

```
-> ipv6 interface vlan_1 retrans-max 5
```

This example sets the maximum number of neighbor solicitations to 5 for the "vlan_1" interface.

To modify the fixed interval at which the neighbor solicitations are sent during the NUD process, use the **ipv6 interface** command with **retrans-timer** parameter. For example:

```
-> ipv6 interface vlan_1 retrans-timer 3000
```

This example sets the interval between neighbor solicitations to 3 seconds (3000 ms) for the "vlan_1" interface.

To enable using an exponentially increasing interval between the neighbor solicitations sent during the NUD process, use the **ipv6 interface** command with **retrans-backoff'** parameter. The interval between subsequent neighbor solicitations will be calculated using the formula, NS interval = ib^n, where,

i = the retransmit interval (retrans-timer)

b = the retransmit backoff (retrans-backoff)

n = the current neighbor solicitation number (starting at 0 for the first transmit)

For example:

```
-> ipv6 interface vlan_1 retrans-max 5 retrans-timer 1000 retrans-backoff 3
```

This example results in up to 5 neighbor solicitations being sent during the NUD process with intervals of 1, 3, 9, and 27 seconds.

Note. The **retrans-max** and **retrans-timer** options also affect neighbor solicitations sent during the initial neighbor discovery process when attempting to resolve a new address. However, the exponential backoff (retrans-backoff) does not apply to initial neighbor discovery.

Configuring Router Advertisement Filtering

To enable Router Advertisement (RA) filtering on an interface, use the **ipv6 ra-filter trusted** command. For example:

```
-> ipv6 ra-filter vlan-3
```

This example enables RA filtering on the “vlan-3” interface. All RAs received on the interface will be dropped.

To specify a trusted port, use the **ipv6 ra-filter trusted** command with the **trusted-port** option. For example:

```
-> ipv6 ra-filter vlan-3 trusted port 1/1/22
```

This specifies that port 1/1/22 is trusted on the “vlan-3” interface. RAs received on this port will be forwarded to all other clients connected to the interface. RAs received on any other port will still be dropped.

To remove a trusted port use the following command:

```
-> no ipv6 ra-filter vlan-3 trusted linkagg 2
```

This will remove linkagg 2 as a trusted port on the “vlan-3” interface.

To disable RA filtering on an interface, use the **no ipv6 ra-filter** command. For example:

```
-> no ipv6 ra-filter vlan-3
```

This disables RA filtering on the vlan-3 interface.

Note. To view the RA filter configuration for an ipv6 interface, use the **show ipv6 ra-filter** command.

Reply or Ignore Echo Requests

By default, the switch will reply to all echo requests, including those sent to anycast or multicast addresses. The **ipv6 echo** command can be used to configure the switch to ignore echo requests sent to anycast or multicast addresses.

Use the **ipv6 echo** command to reply to multicast or any echo requests:

```
-> ipv6 echo multicast

-> ipv6 echo anycast
```

Use no option in **ipv6 echo** command to ignore reply to anycast or multicast echo requests.

```
-> no ipv6 echo anycast
```

ICMPv6 Error Message Rate Limiting

The **ipv6 icmp rate-limit** command configures the rate limit interval and maximum burst size. This command configures the maximum number of ICMPv6 error messages that may be sent in one burst, regardless of the interval, per-VRF basis.

AOS uses the token bucket method for rate limiting ICMPv6 error messages on a per-interface basis. Tokens are added to an interface's bucket at the specified rate limit interval up to the maximum burst value. When an ICMPv6 error message needs to be sent, a token is removed from the bucket. If a token was available the message is sent, otherwise it is discarded.

```
-> ipv6 icmp rate-limit interval 100

-> ipv6 icmp rate-limit burst 20
```

Use **no** option to disable the ICMPv6 error message rate limiting.

```
-> no ipv6 icmp rate-limit
```

The **show ipv6 information** command displays the ICMPv6 rate limit values.

IPv6 EMP Interface

IPv6 EMP interface is an IPv6 interface associated with the physical EMP port.

Only one global unicast address can be assigned to an IPv6 EMP interface. Link-local addresses will be assigned when the interface is first enabled and updated with IPv4 EMP address. If the switch has an EMP port even without the global unique IPv6 address configuration, IPv6 EMP interface shall be created and assigned with the link local address created using the MAC address of the EMP port. Since Duplicate Address Detection (DAD) is disabled on the EMP interface, it is required to configure an unique IPv6 Address that does not conflict with any other IPv6 interface configured in the system.

Only one EMP IPv6 interface may be created per switch in the default VRF. IPv6 EMP address can be added or modified using **modify boot parameters** command. EMP interface is externally reachable through the EMP port after reload.

If an IPv6 address is configured on the EMP port on a VC or chassis setup, ensure to configure the IPv6 address on all the VC elements or CMM.

Configure IPv6 EMP Interface

You must be on the system console to modify the system boot parameters. Use **modify boot parameters** command to configure an IPv6 EMP interface. This enters the **boot** mode. IPv6 EMP interface parameters like the unique IP address and mask, baudrate, priority, and so on can be configured after entering this mode.

It is required to reload the switch for the modified interface configuration to come into effect.

```
-> modify boot parameters
Please wait...
Type '?' for help, 'exit' to exit the boot param parser.
Boot > ?
boot empipaddr         <ip address>
boot empmasklength     <number of bits in mask>
boot serialbaudrate    <1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800,
115200>
boot serialparity      <none, even, odd>
boot serialwordsize    <7, 8>
boot serialstopbits    <1, 2>
boot serialmode        <modemControlOn, modemControlOff>
boot empipv6addr       <ipv6 address>
boot empipv6masklength <number of bits in mask>
'show'           - Display the edit buffer
'commit boot'    - Commit the changes to non volatile memory for future boots
'commit system'  - Commit the changes to running system ONLY
                   Note: EMP changes will only take effect on a future boot
'exit'           - Exit (quit)
```

The **show** command displays the existing configuration of EMP interface.

```
Boot > show
EMP IP Address            : 10.200.105.21/24
Serial (console) baud     : 9600
Serial (console) parity   : none
Serial (console) wordsize : 8
Serial (console) stopbits : 1
Serial (console) mode     : modemControlOff
EMP IPV6 Address          : /
```

To Configure IPv6 EMP interface, configure the IP address and mask:

```
Boot > boot empipv6masklength 64
Boot > show
EMP IP Address            : 10.200.105.21/24
Serial (console) baud     : 9600
Serial (console) parity   : none
Serial (console) wordsize : 8
Serial (console) stopbits : 1
Serial (console) mode     : modemControlOff
EMP IPV6 Address          : /64

Boot > boot empipv6addr  2001::205
Boot > show
EMP IP Address            : 10.200.105.21/24
Serial (console) baud     : 9600
Serial (console) parity   : none
Serial (console) wordsize : 8
Serial (console) stopbits : 1
Serial (console) mode     : modemControlOff
EMP IPV6 Address          : 2001::205/64
Boot >
```

Use the **show ipv6 interface** and **show ipv6 emp-interface** commands to view the EMP interface configuration. The **show ipv6 emp-routes** command displays the IPv6 routes targeted to EMP port/IPv6 EMP interface configuration.

Verifying the IPv6 Configuration

A summary of the show commands used for verifying the IPv6 configuration is given here:

show ipv6 redist	Displays the route map redistribution configuration.
show ipv6 interface	Displays the status and configuration of IPv6 interfaces.
show ipv6 tunnel configured	Displays IPv6 configured tunnel information.
show ipv6 tunnel 6to4	Displays IPv6 6to4 tunnel information.
show ipv6 routes	Displays the IPv6 Forwarding Table.
show ipv6 route-pref	Displays the configured route preference of a router.
show ipv6 router database	Displays a list of all routes (static and dynamic) that exist in the IPv6 router database.
show ipv6 prefixes	Displays IPv6 subnet prefixes used in router advertisements.
show ipv6 neighbors	Displays the IPv6 Neighbor Table.
show ipv6 ra-filter	Displays the RA filter configuration for an IPv6 interface.
show ipv6 tcp listeners	Displays statistics for IPv6 listener traffic.
show ipv6 tcp connections	Displays statistics for IPv6 connection traffic.
show ipv6 icmp statistics	Displays ICMP6 statistics.
show ipv6 pmtu table	Displays the IPv6 Path MTU Table.
show ipv6 tcp connections	Displays TCP Over IPv6 Connection Table. Contains information about existing TCP connections between IPv6 endpoints.
show ipv6 tunnel 6to4	Displays the UDP Over IPv6 Listener Table. Contains information about UDP/IPv6 endpoints.
show ipv6 information	Displays IPv6 information.
show ipv6 emp-interface	Displays the IPv6 EMP interface configuration.
show ipv6 emp-routes	Displays the IPv6 routes targeted to EMP port/IPv6 EMP interface configuration.

For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

20 Configuring IPsec

Internet Protocol security (IPsec) is a suite of protocols for securing IPv6 and IPv4 communications by authenticating and/or encrypting each IPv6 and IPv4 packet in a data stream. IPsec is a framework of open standards designed to provide interoperable, high quality, cryptographically-based security for IPv6 and IPv4 networks through the use of appropriate security protocols, cryptographic algorithms, and cryptographic keys. The set of security services offered includes access control, connectionless integrity, data origin authentication, detection and rejection of replays (a form of partial sequence integrity), and confidentiality (via encryption).

These security services are provided through use of two security protocols, the Authentication Header (AH) and the Encapsulating Security Payload (ESP), and through the use of cryptographic key management procedures and protocols.

In This Chapter

This chapter describes the basic components of IPsec and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- "Configuring an IPsec Master Key" on page 20-10.
- "Configuring an IPsec Policy" on page 20-11.
- "Configuring an IPsec Rule" on page 20-14.
- "Assigning an Action to a Policy" on page 20-13.
- "Configuring an IPsec SA" on page 20-15.
- "Configuring IPsec SA Keys" on page 20-17.
- "Enabling and Disabling Default Discard Policy" on page 20-18.

IPsec Defaults

The following table shows the default settings of the configurable IPsec parameters.

Parameter Description	Command	Default Value/Comments
IPsec global status (A license file must be present on the switch)	N/A	Disabled
Master security key for the switch	**ipsec security-key**	No master security key set
IPsec policy priority	**ipsec policy**	100
IPsec security policy status	**ipsec policy**	Disabled
IPsec discard policy status	**ipsec policy**	Enabled
IPsec SA status	**ipsec sa**	Disabled
Key length AES-CBC	**ipsec sa**	128 bits

Quick Steps for Configuring an IPsec AH Policy

IP Authentication Header (AH) provides data origin authentication, data integrity, and replay protection. Data integrity verifies that the contents of the datagram were not changed in transit, either deliberately or due to random errors, however, AH does not provide data encryption.

1 Configure the master security key. The master security key must be set if keys are to be encrypted when saved in the boot.cfg and snapshot files.

```
-> ipsec security-key master-key-12345
```

2 Define the policy. A policy defines the traffic that requires IPsec protection. The commands below define a bi-directional policy for any protocol and the associated IPv6 address ranges. For example:

```
-> ipsec policy ALLoutMD5 source 664:1:1:1::199/64 destination 664:1:1:1::1/64
protocol any out ipsec admin-state disable

-> ipsec policy ALLinMD5 source 664:1:1:1::1/64 destination 664:1:1:1::199/64
protocol any in ipsec admin-state disable
```

3 Define the rule. A rule defines the security services for the traffic defined by its associated policy. For example the commands below add an AH rule to the polices defined above:

```
-> ipsec policy ALLoutMD5 rule 1 ah
-> ipsec policy ALLinMD5 rule 1 ah
```

4 Enable the policies. A policy cannot be enabled until the rules are defined. Now that rules have been defined, enable the policy using the commands below:

```
-> ipsec policy ALLoutMD5 admin-state enable
-> ipsec policy ALLinMD5 admin-state enable
```

5 Define the Security Keys. Each SA has its own unique set of security keys. The key name is the SA name that is going to use the key and the length must match the authentication algorithm key size. Keys must be defined before the SA can be enabled.

```
-> ipsec key ALLoutMD5_SA sa-authentication  0x11112222333344445555666677778888
-> ipsec key ALLinMD5_SA sa-authentication  0x11112222333344445555666677778888
```

6 Define the SA. An SA specifies the actual actions to be performed. The security parameters index (SPI) helps identify the source/destination pair. The security parameters index (SPI) in combination with the source and destination addresses uniquely identifies an SA. An identical SA (same SPI, source, and destination) must be configured on both systems exchanging IPsec protected traffic.

```
-> ipsec sa ALLoutMD5_SA ah source 664:1:1:1::199 destination 664:1:1:1::1 spi
2000 authentication HMAC-MD5 admin-state enable

-> ipsec sa ALLinMD5_SA ah source 664:1:1:1::1 destination 664:1:1:1::199 spi
2001 authentication HMAC-MD5 admin-state enable
```

7 Use the following show commands to verify the IPsec configuration:

```
-> show ipsec policy
-> show ipsec sa
-> show ipsec key sa-authentication
```

Quick Steps for Configuring an IPsec Discard Policy

IPsec can be used for discarding IPv6 and IPv4 traffic as well as configuring encryption and authentication. For discard policies, no rules, SAs or keys need to be defined.

1 Define the policy. The commands below use similar policy information as in the previous example but the action has been changed to discard:

```
-> ipsec policy Discard_ALLoutMD5 source 664:1:1:1::199/64 destination
664:1:1:1::1/64 protocol any out discard admin-state enable

-> ipsec policy Discard_ALLinMD5 source 664:1:1:1::1/64 destination
664:1:1:1::199/64 protocol any in discard admin-state enable
```

2 Use the following show commands to verify the IPsec configuration:

```
-> show ipsec policy
-> show ipsec statistics
```

IPsec Overview

IPsec provides protection to IPv6 and IPv4 traffic. To achieve this, IPsec provides security services for IPv6 and IPv4 packets at the network layer. These services include access control, data integrity, authentication, protection against replay, and data confidentiality. IPsec enables a system to select the security protocols, encryption and authentication algorithms, and use any cryptographic keys as required. IPsec uses the following two protocols to provide security for an IPv6 and IPv4 datagram:

- Encapsulating Security Payload (ESP) to provide confidentiality, data origin authentication and connectionless integrity.
- Authentication Header (AH) to provide connectionless integrity and data origin authentication for IPv6 and IPv4 datagrams and to provide optional protection against replay attacks. Unlike ESP, AH does not provide confidentiality.

IPsec on an OmniSwitch operates in Transport mode. In transport mode only the payload of the IPv6 and IPv4 packet is encapsulated, and an IPsec header (AH or ESP) is inserted between the original IPv6 or IPv4 header and the upper-layer protocol header. The figure below shows an IPv6 packet protected by IPsec in transport mode.

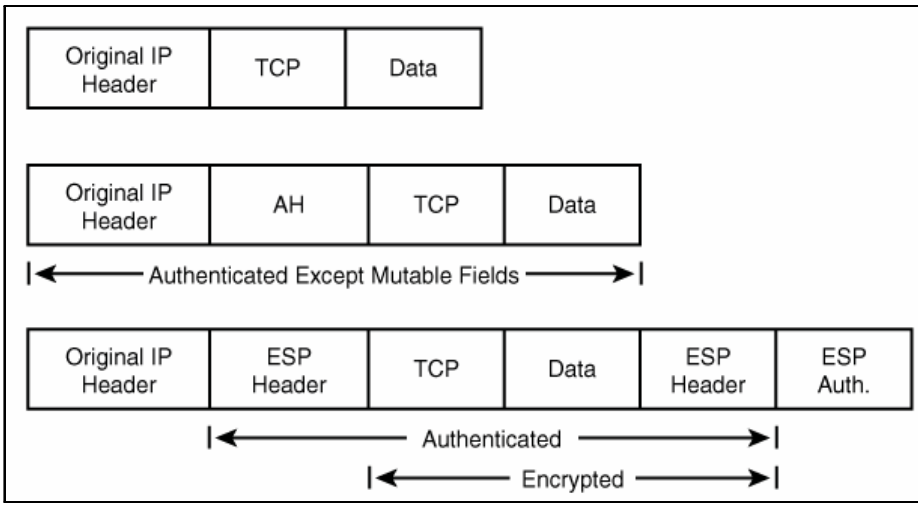


Figure 20-1 : IP Packet in IPsec Transport Mode

Note. The OmniSwitch currently supports the Transport Mode of operation.

Encapsulating Security Payload (ESP)

The ESP protocol provides a means to ensure privacy (encryption), source authentication, and content integrity (authentication). It helps provide enhanced security of the data packet and protects it against eavesdropping during transit.

Unlike AH which only authenticates the data, ESP encrypts data and also optionally authenticates it. It provides these services by encrypting the original payload and encapsulating the packet between a header and a trailer, as shown in the figure below.

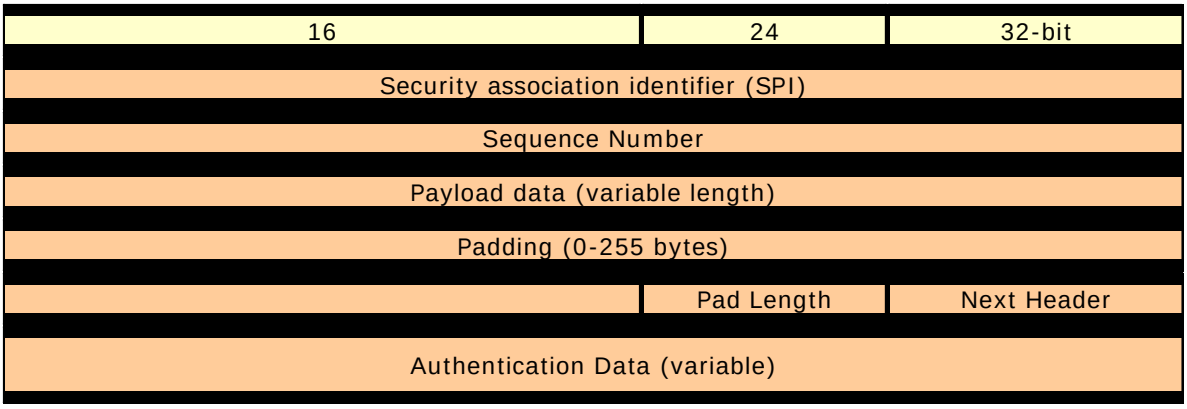


Figure 20-2 : IP Packet protected by ESP

ESP is identified by a value of 50 in the IP header. The ESP header is inserted after the IP header and before the upper layer protocol header. The Security Parameter Index (SPI) in the ESP header is a 32-bit value that, combined with the destination address and protocol in the preceding IP header, identifies the security association (SA) to be used to process the packet. SPI helps distinguish multiple SA's configured for the same source and destination combination. The payload data field carries the data that is being encrypted by ESP. The Authentication digest in the ESP header is used to verify data integrity. Authentication is always applied after encryption, so a check for validity of the data is done upon receipt of the packet and before decryption.

Encryption Algorithms

There are several different encryption algorithms that can be used in IPsec. However, the most commonly used algorithms are "AES" and "3DES". These algorithms are used for encrypting IP packets.

- Advanced Encryption Standard - Cipher Block Chaining - (AES-CBC)

The AES-CBC mode comprises three different key lengths; AES-128, AES-192 and AES-256. Each block of plain text is XOR'd with the previous encrypted block before being encrypted again.

- Triple DES (3DES)

A mode of the DES encryption algorithm that encrypts data three times. Three 64-bit keys are used, instead of one, for an overall key length of 192 bits (the first encryption is encrypted with second key, and the resulting cipher text is again encrypted with a third key). 3DES is a more powerful version of DES.

Authentication Header (AH)

An Authentication Header (AH) provides connectionless integrity and data origin authentication. This protocol permits communicating parties to verify that data was not modified in transit and that it was genuinely transmitted from the apparent source. AH helps verify the authenticity/integrity of the content and origin of a packet. It can optionally protect against replay attacks by using the sliding window technique and discarding old packets. It authenticates the packet by calculating the checksum via hash-based message authentication code (HMAC) using a secret key and either HMAC-MD-5 or HMAC-SHA1 hash functions.

Authentication Algorithms

- HMAC-MD5 - An algorithm that produces a 128-bit hash (also called a digital signature or message digest) from a message of arbitrary length and a 16-byte key. The resulting hash is used, like a fingerprint of the input, to verify content and source authenticity and integrity.
- HMAC-SHA1 - An algorithm that produces a 160-bit hash from a message of arbitrary length and a 20-byte key. It is generally regarded as more secure than MD5 because of the larger hashes it produces.
- AES-XCBC-MAC-96 - An algorithm that uses AES [AES] in CBC mode [MODES] with a set of extensions [XCBC-MAC-1] to overcome the limitations of the classic CBC-MAC algorithm. It uses the AES block cipher with an increased block size and key length (128 bits) which enables it to withstand continuing advances in crypto-analytic techniques and computational capability. Its goal is to ensure that the datagram is authentic and cannot be modified in transit.

Unlike ESP, AH does not encrypt the data. Therefore, it has a much simpler header than ESP. The figure below shows an AH-protected IPv6 packet.

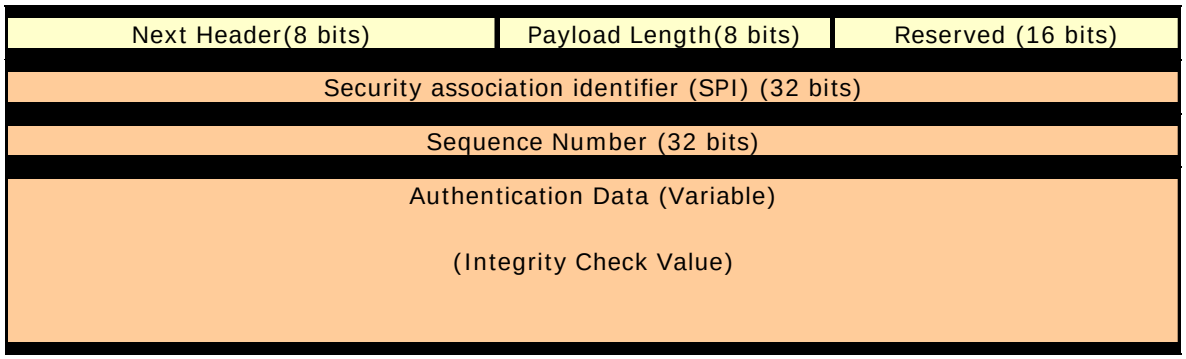


Figure 20-3 : IP Packet protected by AH

AH is identified by a value of 51 in the IPv6 header. The Next header field indicates the value of the upper layer protocol being protected (for example, UDP or TCP) in the transport mode. The payload length field in the AH header indicates the length of the header. The SPI, in combination with the source and destination addresses, helps distinguish multiple SAs configured for the same source and destination combination. The AH header provides a means to verify data integrity. It is similar to the integrity check provided by the ESP header with one key difference. The ESP integrity check only verifies the contents of the ESP payload. AH's integrity check also includes portions of the packet header as well.

IPsec on the OmniSwtich

IPsec allows the following 3 types of actions to be performed on an IPv6 or IPv4 datagram that matches the filters defined in the security policy:

- The IPv6 or IPv4 datagram can be subjected to IPsec processing, i.e. encrypted, and/or authenticated via ESP and AH protocols.
- The IPv6 or IPv4 datagram can be discarded.
- The IPv6 or IPv4 datagram can be permitted to pass without being subjected to any IPsec processing.

The system decides which packets are processed and how they are processed by using the combination of the policy and the SA. The policy is used to specificy which IPsec protocols are used such as AH or ESP while the SA specifies the algorithms such as AES and HMAC-MD5.

Securing Traffic Using IPsec

Securing traffic using IPsec requires the following main procedures below:

- Master Security Key—Used to encrypt SA keys when stored on the switch.
- Policies—Determines which traffic should be processed using IPsec.
- Policy Rules—Determines whether AH, ESP, or a combination of both should be used.
- Security Associations (SAs)—Determines which algorithms should be used to secure the traffic.
- SA Keys—Determines the keys to be used with the SA to secure the traffic.

Master Security Key

The master security key is used to encrypt and decrypt the configured SA keys that are saved to permanent storage (e.g., **boot.cfg** file). If no master security key is configured, SA keys are stored unencrypted. Therefore, configuring a master key is **VITALLY IMPORTANT** and **STRONGLY RECOMMENDED**. A warning message will be logged if the configuration is saved without a Master Security Key being set.

IPsec Policy

IPsec Policies define which traffic requires IPsec processing. The policy requires the source and destination of the traffic to be specified as IPv6 or IPv4 addresses. The policy may cover all traffic from source to destination or may further restrict it by specifying an upper-layer protocol, source, and/or destination ports. Each policy is unidirectional, applying either to inbound or outbound traffic. Therefore, to cover all traffic between a source and destination, two policies would need to be defined.

IPsec Policy Rules

Rules are created and applied to policies. Rules determine what type of encryption or authentication should be used for the associated policy. For example, for a security policy where an IPv6 payload should be protected by an ESP header, which should then be protected by an AH header, two rules would be applied to the policy, one for ESP and one for AH.

Security Association (SA)

A Security Association, more commonly referred to as an SA, is a basic building block of IPsec. It specifies the actual IPsec algorithms to be employed. SA is a unidirectional agreement between the participants regarding the methods and parameters to use in securing a communication channel. A Security Association is a management tool used to enforce a security policy in the IPsec environment. SA actually specifies encryption and authentication between communicating peers.

Manually configured SAs are unidirectional; bi-directional communication requires at least two SAs, one for each direction. Manually-configured SAs are specified by a combination of their SPI, source and destination addresses. However, multiple SAs can be configured for the same source and destination combination. Such SAs are distinguished by a unique Security Parameter Index (SPI).

SA Keys

Keys are used for encrypting and authenticating the traffic. Key lengths must match what is required by the encryption or authentication algorithm specified in the SA. Key values may be specified either in hexadecimal format or as a string.

Note. The OmniSwitch currently supports manually configured SAs only.

Discarding Traffic using IPsec

In order to discard IPv6 or IPv4 datagrams, a policy is configured in the same manner as an IPsec security policy, the difference being that the action is set to ‘discard’ instead of ‘ipsec’. A discard policy can prevent IPv6 or IPv4 traffic from traversing the network.

Configuring IPsec on the OmniSwitch

Before configuring IPsec the following security best practices should be followed:

- Set the Master Security Key—This is used to encrypt SA keys when stored.
- Use SSH, HTTPS, or SNMPv3 to prevent sensitive information such as SA keys from being sent in the clear.
- Restrict IPsec commands to authorized users only. This is described in Chapter 6, "Managing Switch User Accounts." in the *OmniSwitch AOS Release 8 Switch Management Guide*.

Configuring IPsec for securing IPv6 or IPv4 traffic on a switch requires several steps which are explained below:

- Configure the master security key for the switch which is used to encrypt and decrypt the configured SA keys. This is described in "Configuring an IPsec Master Key" on page 20-10.
- Configure an IPsec Security Policy on the switch. This is described in "Configuring an IPsec Policy" on page 20-11.
- Set an IPsec rule for the configured IPsec Security Policy on the switch. This is described in "Configuring an IPsec Rule" on page 20-14.
- Enable the Security Policy. This is described in "Enabling and Disabling a Policy" on page 20-12.
- Configure the authentication and encryption keys required for manually configured IPsec Security associations (SA). This is described in "Configuring IPsec SA Keys" on page 20-17
- Configure an IPsec Security Association on the switch by setting parameters such as Security Association type, encryption and authentication for SA. This is described in "Configuring an IPsec SA" on page 20-15.

Configuring IPsec for discarding IPv6 or IPv4 traffic on a switch requires a single step:

- Configure the IPsec Discard policy on the switch which is used to discard or filter the IPv6 or IPv4 packets. This is described in "Discarding Traffic using IPsec" on page 20-9.

Configuring an IPsec Master Key

The master security key is used to encrypt and decrypt the configured SA keys that are saved to permanent storage (e.g., **boot.cfg** file). To set a master security key the first time, simply enter the **ipsec security-key** command along with a new key value. For example:

```
-> ipsec security-key new_master_key_1
```

or

```
-> ipsec security-key 0x12345678123456781234567812345678
```

Note. The key value can be specified either in hexadecimal format (16 bytes in length) or as a string (16 characters in length). A warning message is logged if SA keys are set without the Master Key being set.

To change the master security key specify the old and new key values.

```
-> ipsec security-key new_master_key_1 new_master_key_2
```

The above command replaces the old security key with the new key value. The old key value must be entered to modify an existing key. If an incorrect old key value is entered, then setting the new key will fail.

When the master security key is set or changed, its value is immediately propagated to the secondary CMM. When the master security key is changed, save and synchronize the current configuration to ensure the proper operation of IPsec in the event of a switch reboot or takeover.

Notes:

- By default, no master security key is set for the switch. When no master security key is configured for the switch, the SA key values are written unencrypted to permanent storage (boot.cfg or other configuration file).
- When running in a virtual chassis setup, the master security key must be manually configured, to the same value, on each switch.

Configuring an IPsec Policy

A policy determines how traffic is going to be processed. For example, policies are used to decide if a particular IPv6 or IPv4 packet needs to be processed by IPsec or not. If security is required, the security policy provides general guidelines as to how it should be provided, and if necessary, links to more specific detail.

Each IPsec security policy is unidirectional and can be applied to IPv6 or IPv4 inbound or outbound traffic depending upon the security level required for the network. Therefore, in order to cover all traffic between source and destination, a minimum of two policies need to be defined; one policy for inbound traffic and another policy for outbound traffic.

To configure an IPsec policy, use the **ipsec policy** command along with the policy name, source address, destination address and optional parameters such as port number, and protocol to which the security policy gets applied. For example:

Local System

```
-> ipsec policy tcp_in source 3ffe:1:1:1::99 destination 3ffe:1:1:1::1 protocol
tcp in ipsec description "IPsec on all inbound TCP" admin-state enable

-> ipsec policy tcp_out source 3ffe:1:1:1::1 destination 3ffe:1:1:1:99 protocol
tcp out ipsec description "IPsec on all outbound TCP" admin-state enable
```

Remote System

```
-> ipsec policy tcp_out source 3ffe:1:1:1::99 destination 3ffe:1:1:1::1 protocol
tcp out ipsec description "IPsec on all outbound TCP" admin-state enable

-> ipsec policy tcp_in source 3ffe:1:1:1::1 destination 3ffe:1:1:1:99 protocol
tcp in ipsec description "IPsec on all inbound TCP" admin-state enable
```

The above commands configure a bi-directional IPsec policy for IPv6 traffic destined to or from the specified IPv6 addresses and indicates the traffic should be processed using IPsec.

Prefixes can also be used when configuring a policy to match a range of addresses as shown below:

```
-> ipsec policy tcp_in source 3ffe::/16 destination 4ffe::/16 protocol tcp in ipsec
description "Any 3ffe to any 4ffe" admin-state enable
```

Use the **no** form of the command to remove the configured IPsec policy. For example:

```
-> no ipsec policy tcp_in
```

Enabling and Disabling a Policy

You can administratively enable or disable the configured security policy by using the keywords **admin-state enable/disable** after the command as shown below:

```
-> ipsec policy tcp_in admin-state disable
```

The above command disables the configured IPsec security policy.

Note. Policies cannot be enabled until at least one rule is configured. See "Configuring an IPsec Rule" on page 20-14.

Assigning a Priority to a Policy

You can use the optional **priority** parameter to assign a priority to the configured IPsec policy so that if IPv6 or IPv4 traffic matches more than one configured policy, the policy with the highest priority is applied to the traffic. The policy with the lower value has the higher priority. For example:

```
-> ipsec policy tcp_in priority 500
```

Note. If two security policies have the same priority then the one configured first will be processed first.

Policy Priority Example

```
-> ipsec policy telnet_deny priority 1000 source ::/0 destination ::/0 port 23
protocol tcp in discard

-> ipsec policy telnet_ipsec priority 200 source 3ffe:1200::/32 destination ::/0
port 23 protocol tcp in ipsec admin-state disable

-> ipsec policy telnet_ipsec rule 1 esp

-> ipsec policy telnet_ipsec admin-state enable

-> ipsec policy telnet_clear priority 100 source 3ffe:1200::1 destination ::/0
port 23 protocol tcp in none

-> ipsec policy telnet_malicious priority 1 source 3ffe:1200::35 destination ::/
0 port 23 protocol tcp in discard
```

1 Policy **telnet_deny** is the lowest priority policy. It will discard any incoming telnet connection attempts.

2 Policy **telnet_ipsec** covers a subset of the source addresses of telnet_deny. With its greater priority, it overrides telnet_deny and allows incoming telnet connections from addresses starting with the prefix 3ffe:1200::/32 as long as they are protected by ESP.

3 The policy **telnet_clear** overrides telnet_ipsec, allowing telnet connection attempts from the host to be accepted without any IPsec protection.

4 Policy **telnet_malicious** can be configured to handle a known malicious system that otherwise would fall under the telnet_ipsec policy. Its priority of 1 ensures that it always takes precedence and discards any incoming telnet connection attempts from the known malicious system.

Assigning an Action to a Policy

To define what action will be performed on the traffic specified in the security policy, you can use the following parameters:

- **discard -** Discards the IP packets.
- **ipsec** - Allows IPsec processing of the traffic to which this policy is applied.

If the action is ipsec, then a rule must be defined before the policy can be enabled. Additionally, SAs and SA keys must also be configured to support the rule.

- **none** - No action is performed.

The above commands could be modified to discard the traffic instead of processing using IPsec.

```
-> ipsec policy tcp_in discard
-> ipsec policy tcp_out discard
```

Configuring the Protocol for a Policy

You can define the type of protocol to which the security policy can be applied by using the **protocol** parameter. For example:

```
-> ipsec policy udp_in source ::/0 destination 3ffe:200:200:4001::99 protocol
udp in ipsec description "IPsec on all inbound UDP" admin-state enable
```

The following table lists the various protocols that can be specified, refer to the **ipsec policy** command for additional details.

protocol			
any	**icmp6[type** *type***]**	**tcp**	**udp**
ospf	**vrrp**	**number** *protocol*	

Verifying a Policy

To verify the configured IPsec policy, use the **show ipsec policy** command. For example:

```
-> show ipsec policy

IPV4 Default discard policy = Disabled
IPV6 Default discard policy = Enabled

Name        Priority Source-> Destination         Protocol Direction Action State
----------+--------+----------------------------+--------+-------+-------+------
tcp_in       500     3ffe:1:1:1::99->3ffe:1:1:1::1   TCP   in     ipsec esp  active
tcp_out      500     3ffe:1:1:1::1->3ffe:1:1:1::99   TCP   out    ipsec esp  active
ftp-in-drop  100    ::/0->::/0                       TCP   in     discard disabled
telnet-in-1  100    2000::/48->::/0                  TCP   in     ipsec   disabled
```

The above command provides examples of various configured policies.

Note. The presence of a '+' sign in the 'Source->Destination' or 'Action' indicates the values has been truncated to fit. View a specific security policy to view additional details.

The policy information can also be viewed based on the address type.

To view the policy information for IPv4 address, enter:

```
-> show ipsec policy ipv4
Default ipv4 discard policy = Disabled

Name     Priority Source -> Destination            Protocol   Direction  Action       State
--------+--------+-------------------------------+----------+---------+-------------+------
udp_in   100      10.1.1.2->10.1.1.1                UDP       in        ipsec ah     active
udp_out  100      10.1.1.1->10.1.1.2                UDP       out        ipsec ah    active
```

To view the policy information for IPv6 address, enter:

```
-> show ipsec policy ipv6
Default ipv6 discard policy = Disabled

Name      Priority Source -> Destination     Protocol     Direction  Action        State
---------+--------+-------------------------+------------+----------+---------+--------------
tcp_in    100      3ffe::200 -> 3ffe::100     TCP          in        ipsec esp     active
tcp_out   100      3ffe::100 -> 3ffe::200     TCP          out       ipsec esp     active
```

You can also verify the configuration of a specific security policy by using the **show ipsec policy** command followed by the name of the security policy. For example:

```
-> show ipsec policy tcp_in
Name        = tcp_in
Priority    = 500
Source      = 3ffe:1:1:1::99
Destination = 3ffe:1:1:1::1
Protocol    = TCP
Direction   = in
Action      = ipsec
State       = active
Rules:
  1 : esp
Description:
  IPsec on all inbound TCP
```

Configuring an IPsec Rule

To configure an IPsec rule for a configured IPsec security policy, use the **ipsec policy rule** command along with the policy name, index value for the IPsec policy rule, and IPsec protocol type (AH or ESP). For example:

```
-> ipsec policy tcp_in rule 1 esp
```

The above command applies the configured IPsec security policy with rule 1 to ESP. The index value specified determines the order in which a rule should get applied to the payload. The policy name configured for the IPsec policy rule should be the same as the policy name configured for the IPsec security policy. It's possible to first encrypt the original content of an IPv6 or IPv4 packet using ESP and then authenticate the packet using AH by configuring an ESP rule with an index of one and then configuring the AH rule with an index of two. For example:

```
-> ipsec policy tcp_in rule 1 esp
-> ipsec policy tcp_in rule 2 ah
```

Use the **no** form of this command to remove the configured IPsec rule for an IPsec security policy. For example:

```
-> no ipsec policy tcp_in rule 2
```

Verifying IPsec rule for IPsec Policy

To verify the IPsec policy, use the **show ipsec policy** command. For example:

```
-> show ipsec policy tcp_in
Name        = tcp_in
Priority    = 500
Source      = 3ffe:1:1:1::99
Destination = 3ffe:1:1:1::1
Protocol    = TCP
Direction   = in
Action      = ipsec
State       = active
Rules:
  1 : esp,
  2 : ah
Description:
  IPsec on all inbound TCP
```

Configuring an IPsec SA

IPsec Security Association (SA) is a set of security information that describes a particular kind of secure connection between two devices. An SA specifies the actual IPsec algorithms applied to the IP traffic (e.g. encryption using 3DES, HMAC-SHA1 for authentication).

To configure an IPsec Security Association, use the **ipsec sa** command along with the type of security association, source address, destination address, encryption and authentication algorithms used for SA. For example:

Local System

```
-> ipsec sa tcp_in_ah ah source 3ffe:1:1:1::99 destination 3ffe:1:1:1::1 spi
9901 authentication hmac-sha1 description "HMAC SHA1 on traffic from 99 to 1"
-> ipsec sa tcp_out_ah ah source 3ffe:1:1:1::1 destination 3ffe:1:1:1::99 spi
9902 authentication hmac-sha1 description "HMAC SHA1 on traffic from 1 to 99"
```

Remote System

```
-> ipsec sa tcp_out_ah ah source 3ffe:1:1:1::99 destination 3ffe:1:1:1::1 spi
9901 authentication hmac-sha1 description "HMAC SHA1 on traffic from 99 to 1"
-> ipsec sa tcp_in_ah ah source 3ffe:1:1:1::1 destination 3ffe:1:1:1::99 spi
9902 authentication hmac-sha1 description "HMAC SHA1 on traffic from 1 to 99"
```

The above commands configure bi-directional IPsec SAs of AH type for data traffic to and from source IPv6 addresses 3ffe:1:1:1::99 and 3ffe:1:1:1::1 with security parameter indexes (SPI) of 9901 and 9902. The combination of SPI, source, and destination addresses uniquely identify an SA. The above commands also configure hmac-shal as the type of authentication algorithm which is to be used for the IPv6 traffic covered by the configured SA.

Note. The IPsec endpoints must have identical SAs (SPI, source address, destination addresses) configured.

Use the **admin-state enable/disable** parameters to enable or disable the SA.

```
-> ipsec sa tcp_in_ah admin-state enable
```

Use the **no** form of the command to disable the SA.

```
-> no ipsec sa tcp_in_ah
```

Configuring ESP or AH

The IPsec SA can be configured as ESP or AH. In the above example, the IPsec SA is configured as AH. You can also configure the SA as ESP, as shown below:

```
-> ipsec sa tcp_in_ah esp source 3ffe:1:1:1::99 destination 3ffe:1:1:1::1 spi
9901 encryption 3DES-CBC description "3DES on traffic from 99 to 1"
```

You can use the **encryption** parameter to specify the encryption algorithm to be used for the traffic covered by the SA. This parameter can only be used when the SA type is ESP.

Configuring the ESP Key Size

Some types of encryption algorithms allow the key size to specified; specifying the key lengths overrides their default values. To do so, use the **key-size** option after the specified encryption algorithm. For example:

```
-> ipsec sa tcp_in_ah esp source 3ffe:1:1:1::99 destination 3ffe:1:1:1::1 spi
9901 encryption aes-cbc key-size 192
```

The above command configures an IPsec SA of ESP using aes-cbs and a key length of 192 bits. You can allow an IPsec SA to operate as an ESP confidentiality-only SA by using the **none** option with the authentication parameter or by simply omitting the authentication parameter from the command.

Refer to "Configuring IPsec SA Keys" on page 20-17 or the **ipsec sa** command for supported encryption types and key lengths.

Verifying IPsec SA

To display the configured IPsec SA, use the **show ipsec sa** command. For example:

```
-> show ipsec sa
Name      Type  Source-> Destination[SPI]           Encryption Authentication State
---------+---+----------------------------------------+----------+-------------+---
tcp_in_ah  ah   3ffe:1:1:1::99 -> 3ffe:1:1:1::1 [9901]  none       hmac-sha1     active
tcp_out_ah ah   3ffe:1:1:1::1 -> 3ffe:1:1:1::99 [9902]  none       hmac-sha1     active
```

To display the configuration of a specific IPsec SA, use the **show ipsec sa** command followed by the name of the configured IPsec SA. For example:

```
-> show ipsec sa tcp_in_ah

Name            = tcp_in_ah
Type            = AH
Source          = 3ffe:1:1:1::99,
Destination     = 3ffe:1:1:1::1,
SPI             = 9901
Encryption      = none
Authentication  = hmac-sha1
State           = active
Description:
  "HMAC SHA1 on traffic from 99 to 1
```

Configuring IPsec SA Keys

To configure the authentication and encryption keys for a manually configured SA, use the **ipsec key** command along with the SA name and key value which will be used for AH or ESP. For example:

```
-> ipsec key tcp_in_ah sa-authentication 0x112233445566778899001122334455 66
```

The above command configures an IPsec SA key named tcp_in_ah. This IPsec SA key will be used for the AH authentication protocol and has a value of 0x112233445566778899001122334455 66.

The length of the key value must match the value that is required by the encryption or authentication algorithm that will use the key. The table shown below displays the key lengths for the supported algorithms:

Algorithm	Key Length
3DES-CBC	192 Bits
AES-CBC	128,192, or 256 Bits
HMAC-MD5	128 Bits
HMAC-SHA1	160 Bits
HMAC-SHA2 -256	256 Bits
HMAC-SHA2-384	384 Bits
HMAC-SHA2-512	512 Bits
AES-XCBC-MAC	128 Bits

Use the following information to determine how to create the proper key size:

- Number of Characters = Key Size (in bits) / 8; Ex. A 160-bit key would require 20 characters for the key.
- Number of Hexidecimal = Key Size (in bits) / 4; Ex. A 160-bit key would require 40 hexidecimal digits.

Note. The *name* parameter must be the same as the name of the manually configured IPsec SA. Also, the combination of the key name and type must be unique.

Use the **no** form of this command to delete the configured IPsec SA key. For example:

```
-> no ipsec key tcp_in_ah
```

Verifying IPsec SA Key

To display the encryption key values which are configured for manually configured IPsec SAs, use the **show ipsec key** command For example:

```
-> show ipsec key sa-encryption
Encryption Keys
Name                  Length (bits)
--------------------+---------------
sa_1                          192
sa_2                          160
sa_3                          64
```

The above command shows the number of manually configured SAs along with their encryption key lengths in bits respectively. To display the IPsec SA keys used for authentication, use the **show ipsec key** command, as shown below:

```
-> show ipsec key sa-authentication
Authentication Keys
Name                    Length (bits)
--------------------+----------------
tcp_in_ah                     160
sa_1                          128
sa_5                          160
```

The above command shows the number of manually configured SAs along with their authentication key lengths in bits respectively.

Note. Due to security reasons, key values will not be displayed; only key names and key lengths will be displayed.

Once IPsec is configured for IPv6 or IPv4 on the switch, you can monitor the incoming and outgoing packets for the configured parameters by using the **show ipsec statistics** command.

```
-> show ipsec statistics

Inbound
  Discarded                   = 0,
  No SA found                 = 0,
  Policy violation            = 0,
  Authentication failure      = 0,
  Replay check failed         = 0,
  Other error                 = 0
Outbound
  Discarded                   = 0,
  No SA found                 = 0,
  Other error                 = 0
```

Enabling and Disabling Default Discard Policy

A default discard IPsec policy drops all the inbound traffic that does not match an IPsec policy. This policy on its own drops all the incoming traffic destined for the switch, hence, it is required to add appropriate higher priority policies to allow the desired traffic to be received. The default discard policy is not applied to the forwarded traffic. The default discard policy can be specifically configured for IPv4 or IPv6 policy.

To enable default discard IPsec policy, use the **enable** keyword:

```
-> ipsec default-discard ipv6 admin-state enable
```

To disable default discard IPsec policy, use the **disable** keyword:

```
-> ipsec default-discard ipv6 admin-state disable
```

The default discard policy on its own drops all the incoming traffic destined for the switch. It is required to add appropriate higher priority policies to allow the desired traffic to be received. At a minimum, policies must be added to allow neighbor discovery traffic to be accepted. For example:

```
-> ipsec policy ns-in priority 100 source ::/0 destination ::/0 protocol ICMP6
type 135 in none

-> ipsec policy na-in priority 100 source ::/0 destination ::/0 protocol ICMP6
type 136 in none
```

The **show ipsec policy** command output display indicates whether the default discard policy is enabled or disabled.

Additional Examples

Configuring ESP

The example below shows the commands for configuring ESP between two OmniSwitches for all TCP traffic.

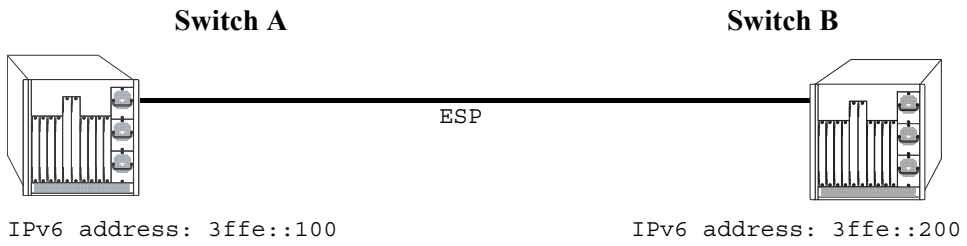


Figure 20-4 : ESP Between Two OmniSwitches

Switch A

```
-> ipsec security-key master-key-12345

-> ipsec policy tcp_out source 3ffe::100 destination 3ffe::200 protocol tcp out
ipsec description "IPsec on TCP to 200"

-> ipsec policy tcp_in source 3ffe::200 destination 3ffe::100 protocol tcp in
ipsec description "IPsec on TCP from 200"

-> ipsec policy tcp_out rule 1 esp

-> ipsec policy tcp_in rule 1 esp

-> ipsec policy tcp_out admin-state enable

-> ipsec policy tcp_in admin-state enable

-> ipsec sa tcp_out_esp esp source 3ffe::100 destination 3ffe::200 spi 1000
encryption des-cbc authentication hmac-sha1 description "ESP to 200" admin-state
enable

-> ipsec sa tcp_in_esp esp source 3ffe::200 destination 3ffe::100 spi 1001
encryption des-cbc authentication hmac-sha1 description "ESP from 200" admin-
state enable

-> ipsec key tcp_out_esp sa-encryption 12345678

-> ipsec key tcp_out_esp sa-authentication 12345678901234567890

-> ipsec key tcp_in_esp sa-encryption 12345678

-> ipsec key tcp_in_esp sa-authentication 12345678901234567890
```

Switch B

```
-> ipsec security-key master-key-12345

-> ipsec policy tcp_out source 3ffe::200 destination 3ffe::100 protocol tcp out
ipsec description "IPsec on TCP to 100"

-> ipsec policy tcp_in source 3ffe::100 destination 3ffe::200 protocol tcp in
ipsec description "IPsec on TCP from 100"

-> ipsec policy tcp_out rule 1 esp

-> ipsec policy tcp_in rule 1 esp

-> ipsec policy tcp_out admin-state enable

-> ipsec policy tcp_in admin-state enable

-> ipsec sa tcp_out_esp esp source 3ffe::200 destination 3ffe::100 spi 1001
encryption des-cbc authentication hmac-sha1 description "ESP to 100" admin-state
enable

-> ipsec sa tcp_in_esp esp source 3ffe::100 destination 3ffe::200 spi 1000
encryption des-cbc authentication hmac-sha1 description "ESP from 100" admin-
state enable

-> ipsec key tcp_out_esp sa-encryption 12345678

-> ipsec key tcp_out_esp sa-authentication 12345678901234567890

-> ipsec key tcp_in_esp sa-encryption 12345678

-> ipsec key tcp_in_esp sa-authentication 12345678901234567890
```

Discarding RIPng Packets

RIPng uses the well known address of ff02::9 to advertise routes. The following example shows how IPsec can be configured to drop all RIPng packets.

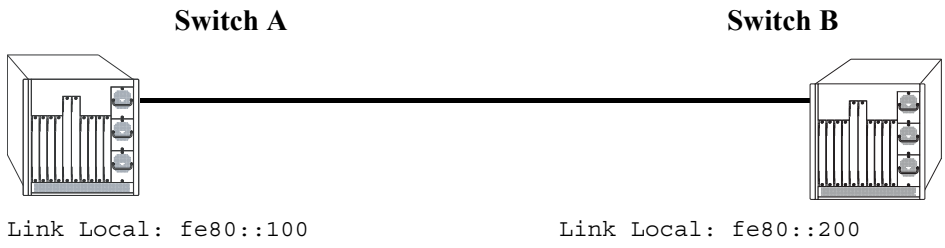


Figure 20-5 : Discarding RIPng Packets

Switch A

```
-> ipsec policy DISCARD_UDPout source fe80::100 destination ff02::9 protocol udp
out discard

-> ipsec policy DISCARD_UDPin source fe80::200 destination ff02::9 protocol udp
in discard
```

Switch B

```
-> ipsec policy DISCARD_UDPout source fe80::200 destination ff02::9 protocol udp
out discard

-> ipsec policy DISCARD_UDPin source fe80::100 destination ff02::9 protocol udp
in discard
```

Verifying the IPsec Configuration

To display information such as details about manually configured IPsec Security Associations and other IPsec parameters configured on the switch, use the **show** commands listed in the following table::

show ipsec sa	Displays information about manually configured IPsec SAs.
show ipsec key	Displays encryption and authentication key values for the manually configured IPsec SA.
show ipsec policy	Displays information about IPsec Security Policies configured for the switch.
show ipsec statistics	Displays IPsec statistics for IPv6 and IPv4 traffic.

For more information about the resulting displays form these commands, see the "IPsec Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Examples of the above commands and their outputs are given in the section "Configuring IPsec on the OmniSwitch" on page 20-10.

21 Configuring RIP

Routing Information Protocol (RIP) is a widely used Interior Gateway Protocol (IGP) that uses hop count as its routing metric. RIP-enabled routers update neighboring routers by transmitting a copy of their own routing table. The RIP routing table uses the most efficient route to a destination, that is, the route with the fewest hops and longest matching prefix.

The switch supports RIP version 1 (RIPv1), RIP version 2 (RIPv2), and RIPv2 that is compatible with RIPv1. It also supports text key and MD5 authentication, on an interface basis, for RIPv2.

In This Chapter

This chapter describes RIP and how to configure it through the Command Line Interface (CLI). It includes instructions for configuring basic RIP routing and fine-tuning RIP by using optional RIP configuration parameters (e.g., RIP send/receive option and RIP interface metric). It also details RIP redistribution, which allows a RIP network to exchange routing information with networks running different protocols (e.g., OSPF and BGP). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

This chapter provides an overview of RIP and includes information about the following procedures:

- RIP Routing
  - Loading RIP (see page 21-6)
  - Enabling RIP (see page 21-7)
  - Creating a RIP Interface (see page 21-7)
  - Enabling a RIP Interface (see page 21-7)
- RIP Options
  - Configuring the RIP Forced Hold-Down Interval (see page 21-9)
  - Configuring the RIP Update Interval (see page 21-9)
  - Configuring the RIP Invalid Timer (see page 21-10)
  - Configuring the RIP Garbage Timer (see page 21-10)
  - Configuring the RIP Hold-Down Timer (see page 21-10)
  - Enabling a RIP Host Route (see page 21-11)
- RIP Redistribution
  - Configuring Route Redistribution (see page page 21-12)
- RIP Security
  - Configuring Authentication Type (see page 21-18)
  - Configuring Passwords (see page 21-18)

RIP Defaults

The following table lists the defaults for RIP configuration through the **ip rip** command.

Description	Command	Default
RIP Status	**ip rip admin-state**	disable
RIP Forced Hold-Down Interval	**ip rip force-holddowntimer**	0
RIP Update Interval	**ip rip update-interval**	30 seconds
RIP Invalid Timer	**ip rip invalid-timer**	180 seconds
RIP Garbage Timer	**ip rip garbage-timer**	120 seconds
RIP Hold-Down Timer	**ip rip holddown-timer**	0
RIP Interface Metric	**ip rip interface metric**	1
RIP Interface Send Version	**ip rip interface send-version**	v2
RIP Interface Receive Version	**ip rip interface recv-version**	both
RIP Host Route	**ip rip host-route**	enable
RIP Route Tag	**ip rip host-route**	0

Quick Steps for Configuring RIP Routing

To forward packets to a device on a different VLAN, you must create a router interface on each VLAN. To route packets by using RIP, you must enable RIP and create a RIP interface on the router interface. The following steps show you how to enable RIP routing between VLANs "from scratch". If active VLANs and router ports have already been created on the switch, go to Step 7.

1 Create VLAN 1 with a description (e.g., VLAN 1) by using the **vlan** command. For example:

```
-> vlan 1 name "VLAN 1"
```

2 Create VLAN 2 with a description (e.g., VLAN 2) by using the **vlan** command. For example:

```
-> vlan 2 name "VLAN 2"
```

3 Assign an active port to VLAN 1 by using the **vlan members untagged** command. For example, the following command assigns port 1 on slot 1 to VLAN 1:

```
-> vlan 1 members port 1/1 untagged
```

4 Assign an active port to VLAN 2 by using the **vlan members** command. For example, the following command assigns port 2 on slot 1 to VLAN 2:

```
-> vlan 2 members port 1/2 untagged
```

5 Configure an IP interface to enable IP routing on a VLAN by using the **ip interface** command. For example:

```
-> ip interface vlan-1 address 171.10.1.1 vlan 1
```

6 Configure an IP interface to enable IP routing on a VLAN by using the **ip interface** command. For example:

```
-> ip interface vlan-2 address 171.11.1.1 vlan 2
```

7 Load RIP into the switch memory by using the **ip load rip** command. For example:

```
-> ip load rip
```

8 Enable RIP on the switch by using the **ip rip admin-state** command. For example:

```
-> ip rip admin-state enable
```

9 Create a RIP interface on VLAN 1 by using the **ip rip interface** command. For example:

```
-> ip rip interface vlan-1
```

10 Enable the RIP interface by using the **ip rip interface admin-state** command. For example:

```
-> ip rip interface vlan-1 admin-state enable
```

11 Create an RIP interface on VLAN 2 by using the **ip rip interface** command. For example:

```
-> ip rip interface vlan-2
```

Note. For more information on VLANs and router ports, see Chapter 4, "Configuring VLANs."

RIP Overview

In switching, traffic can be transmitted from one media type to another within the same VLAN. Switching happens at Layer 2, the link layer; routing happens at Layer 3, the network layer. In IP routing, traffic can be transmitted across VLANs. When IP routing is enabled, the switch uses routing protocols to build routing tables that keep track of stations in the network and decide the best path for forwarding data. When the switch receives a packet to be routed, it strips off the MAC header and examines the IP header. It looks up the source/destination address in the routing table, and then adds the appropriate MAC address to the packet. Calculating routing tables and stripping/adding MAC headers to packets is performed by switch software.

IP is associated with several Layer 3 routing protocols. RIP is built into the base code loaded onto the switch. Others are part of the optional OmniSwitch Advanced Routing Software implementation. IP supports the following IP routing protocols:

- RIP—An IGP that defines how routers exchange information. RIP makes routing decisions by using a "least-cost path" method. RIPv1 and RIPv2 services allow the switch to learn routing information from neighboring RIP routers. For more information and instructions for configuring RIP, see "RIP Routing" on page 21-6.
- Open Shortest Path First (OSPF)—An IGP that provides a routing function similar to RIP but uses different techniques to determine the best route for a datagram. OSPF is part of the optional Advanced Routing Software. For more information see the "Configuring OSPF" chapter in the *OmniSwitch AOS Release 8 Advanced Routing Configuration Guide*.

When RIP is initially enabled on a switch, it issues a request for routing information, and listens for responses to the request. If a switch configured to supply RIP hears the request, it responds with a response packet based on information in its routing database. The response packet contains destination network addresses and the routing metric for each destination. When a RIP response packet is received, RIP takes the information and rebuilds the switch's routing database, adding new routes and "better" (lower metric) routes to destinations already listed in the database.

RIP uses a hop count metric to measure the distance to a destination. In the RIP metric, a switch advertises directly connected networks at a metric of 1. Networks that are reachable through one other gateway are 2 hops, networks that are reachable through two gateways are 3 hops, etc. Thus, the number of hops (or hop count) along a path from a given source to a given destination refers to the number of networks that are traversed by a datagram along that path. When a switch receives a routing update that contains a new or changed destination network entry, the switch adds one to the metric value indicated in the update and enters the network in the routing table. After updating its routing table, the switch immediately begins transmitting routing updates to inform other network switches of the change. These updates are sent independently of the regularly scheduled updates. By default, RIP packets are broadcast every 30 seconds, even if no change has occurred anywhere in a route or service.

RIP deletes routes from the database if the next switch to that destination says the route contains more than 15 hops. In addition, all routes through a gateway are deleted by RIP if no updates are received from that gateway for a specified time period. If a gateway is not heard from for 120 seconds, all routes from that gateway are placed in a hold-down state. If the hold-down timer value is exceeded, the routes are deleted from the routing database. These intervals also apply to deletion of specific routes.

RIP Version 2

RIP version 2 (RIPv2) adds additional capabilities to RIP. Not all RIPv2 enhancements are compatible with RIPv1. To avoid supplying information to RIPv1 routes that could be misinterpreted, RIPv2 can only use non-compatible features when its packets are multicast. Multicast is not supported by RIPv1. On interfaces that are not compatible with IP multicast, the RIPv1-compatible packets used do not contain potentially confusing information. RIPv2 enhancements are listed below.

- Next Hop—RIPv2 can advertise a next hop other than the switch supplying the routing update. This capability is useful when advertising a static route to a silent switch not using RIP, since packets passing through the silent switch do not have to cross the network twice.
- Network Mask—RIPv1 assumes that all subnetworks of a given network have the same network mask. It uses this assumption to calculate the network masks for all routes received. This assumption prevents subnets with different netmasks from being included in RIP packets. RIPv2 adds the ability to specify the network mask with each network in a packet. Because RIPv1 switches ignore the network mask in RIPv2 packets, their calculation of the network mask could possibly be wrong. For this reason, RIPv1-compatible RIPv2 packets cannot contain networks that would be misinterpreted by RIPv1. These networks must only be provided in native RIPv2 packets that are multicast.
- Authentication—RIPv2 packets can contain an authentication key that can be used to verify the validity of the supplied routing data. Authentication can be used in RIPv1-compatible RIPv2 packets, but RIPv1 switches ignore authentication information. Authentication is a simple password in which an authentication key of up to 16 characters is included in the packet. If this key does not match the configured authentication key, the packet is discarded. For more information on RIP authentication, see "RIP Security" on page 21-18.
- IP Multicast—IP Multicast Switching (IPMS) is a one-to-many communication technique employed by emerging applications such as video distribution, news feeds, netcasting, and resource discovery. Unlike unicast, which sends one packet per destination, multicast sends one packet to all devices in any subnetwork that has at least one device requesting the multicast traffic. For more information on IPMS, see Chapter 27, "Configuring IP Multicast Switching."

RIP Routing

IP routing requires IP router interfaces to be configured on VLANs and a routing protocol to be enabled and configured on the switch. RIP also requires a RIP interface to be created and enabled on the routing interface. In the illustration below, a router interface and RIP interface have been configured on each VLAN. Therefore, workstations connected to ports on VLAN 1 on Switch 1 can communicate with VLAN 2; workstations connected to ports on VLAN 3 on Switch 2 can communicate with VLAN 2. Also, ports from both switches have been assigned to VLAN 2, and a physical connection has been made between the switches. Therefore, workstations connected to VLAN 1 on Switch 1 can communicate with workstations connected to VLAN 3 on Switch 2.

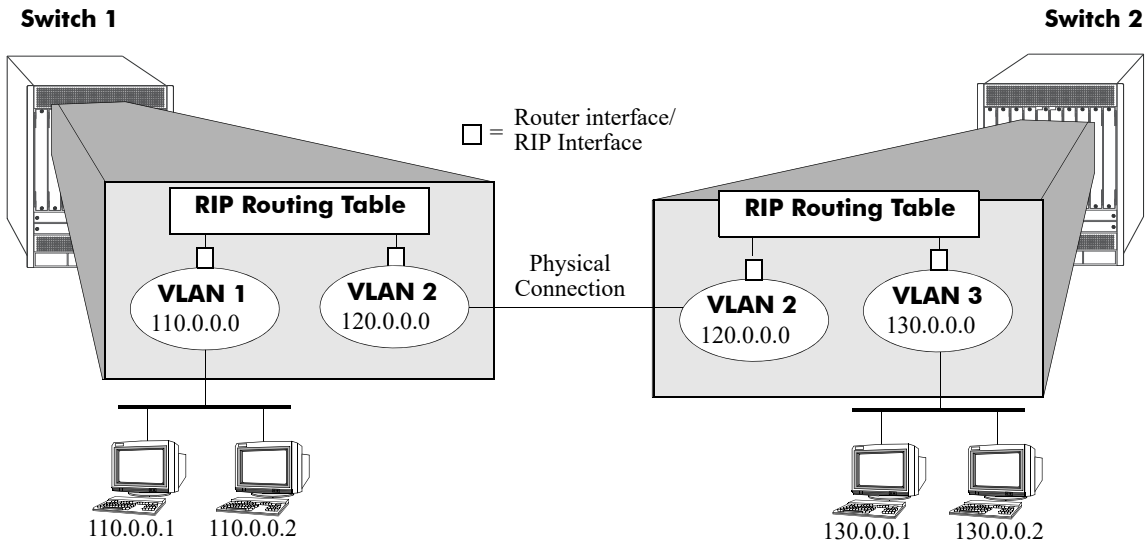


Figure 21-1 : RIP Routing

Loading RIP

When the switch is initially configured, RIP must be loaded into the switch memory. Use the **ip load rip** command to load RIP.

To remove RIP from the switch memory, you must manually edit the **boot.cfg** file. The **boot.cfg** file is an ASCII text-based file that controls many of the switch parameters. Open the file and delete all references to RIP. You must reboot the switch when this is complete.

Note. In simple networks where only IP forwarding is required, you need not use RIP. If you are not using RIP, it is best not to load it to save switch resources.

Enabling RIP

RIP is disabled by default. Use the **ip rip admin-state** command to enable RIP routing on the switch. For example:

```
-> ip rip admin-state enable
```

Use the **ip rip admin-state disable** command to disable RIP routing on the switch. Use the **show ip rip** command to display the current RIP status.

Creating a RIP Interface

You must create a RIP interface on a VLAN's IP router interface to enable RIP routing. Enter the **ip rip interface** command followed by the name of the VLAN router port. For example, to create a RIP interface on a router port with a name of rip-1 you would enter:

```
-> ip rip interface rip-1
```

Use the **no ip rip interface** command to delete a RIP interface. Use the **show ip rip interface** command to display configuration and error information for a RIP interface.

Note. You can create a RIP interface even if an IP router interface has not been configured. However, RIP does not function unless a RIP interface is created and enabled on an IP router interface. See Chapter 4, "Configuring VLANs," and Chapter 17, "Configuring IP," for more information.

Enabling a RIP Interface

Once you have created a RIP interface, you must enable it to enable RIP routing. Use the **ip rip interface admin-state** command followed by the interface IP address to enable a RIP interface. For example, to enable RIP routing on a RIP interface rip-1 you would enter:

```
-> ip rip interface rip-1 admin-state enable
```

To disable an RIP interface, use the **disable** keyword with the **ip rip interface admin-state** command. For example to disable RIP routing on a RIP interface rip-1 you would enter:

```
-> ip rip interface rip-1 admin-state disable
```

Configuring the RIP Interface Send Option

The RIP Send option defines the type(s) of RIP packets that the interface sends. Using this command overrides RIP default behavior. Other devices must be able to interpret the information provided by this command or routing information is not properly exchanged between the switch and other devices on the network.

Use the **ip rip interface send-version** command to configure an individual RIP interface Send option. Enter the IP address of the RIP interface, and then enter a Send option. For example, to configure a RIP interface rip-1 to send only RIPv1 packets you would enter:

```
-> ip rip interface rip-1 send-version v1
```

The Send options are:

- **v1.** Only RIPv1 packets is sent by the switch.

- **v2.** Only RIPv2 packets is sent by the switch.
- **v1compatible.** Only RIPv2 broadcast packets (not multicast) is sent by the switch.
- **none.** Interface does not forward RIP packets.

To set the default RIP send option use the **ip rip interface send-version** command.

Use the **show ip rip interface** command to display the current interface send option.

Configuring the RIP Interface Receive Option

The RIP Receive option defines the type(s) of RIP packets that the interface accepts. Using this command overrides RIP default behavior. Other devices must be able to interpret the information provided by this command or routing information is not properly exchanged between the switch and other devices on the network.

Use the **ip rip interface recv-version** command to configure an individual RIP interface Receive option. Enter the IP address of the RIP interface, and then enter a Receive option. For example, to configure RIP interface rip-1 to receive only RIPv1 packets you would enter:

```
-> ip rip interface rip-1 recv-version v1
```

The Receive options are:

- **v1.** Only RIPv1 packets is received by the switch.
- **v2.** Only RIPv2 packets is received by the switch.
- **both.** Both RIPv1 and RIPv2 packets is received by the switch.
- **none.** Interface ignores any RIP packets received.

To set the default RIP receive option use the **ip rip interface recv-version** command.

Configuring the RIP Interface Metric

You can set priorities for routes generated by a switch by assigning a metric value to routes generated by that switch's RIP interface. For example, routes generated by a neighboring switch can have a hop count of 1. However, you can lower the priority of routes generated by that switch by increasing the metric value for routes generated by the RIP interface.

Note. When you configure a metric for a RIP interface, this metric cost is added to the metric of the incoming route.

Use the **ip rip interface metric** command to configure the RIP metric or cost for routes generated by a RIP interface. Enter the IP address of the RIP interface as well as a metric value. For example, to set a metric value of 2 for the RIP interface rip-1 you would enter:

```
-> ip rip interface rip-1 metric 2
```

The valid metric range is **1** to **15**. To change the default value use the **ip rip interface metric** command.

Use the **show ip rip interface** command to display the current interface metric.

Configuring the RIP Interface Route Tag

Use the **ip rip route-tag** command to configure a route tag value for routes generated by the RIP interface. This value is used to set priorities for RIP routing. Enter the command and the route tag value. For example, to set a route tag value of 1 you would enter:

```
-> ip rip route-tag 1
```

The valid route tag value range is **1** to **2147483647**.

Use the **show ip rip** command to display the current route tag value.

RIP Options

The following sections detail procedures for configuring RIP options. RIP must be loaded and enabled on the switch before you can configure any of the RIP configuration options.

Configuring the RIP Forced Hold-Down Interval

The RIP forced hold-down timer value defines an amount of time, in seconds, during which routing information regarding better paths is suppressed. A route enters into a forced hold-down state when an update packet is received that indicates the route is unreachable and when this timer is set to a non-zero value. After this timer has expired and if the value is less that 120 seconds, the route enters a hold-down state for the rest of the period until the remainder of the 120 seconds has also expired. During this time the switch accepts any advertisements for better paths that are received.

Note that the RIP forced hold-down timer is *not* the same as the RIP hold-down timer. The forced hold-down timer defines a separate interval that overlaps the hold-down state. During the forced hold-down timer interval, the switch does not accept *better* routes from other gateways. For more information on RIP hold-down timer, see "Configuring the RIP Hold-Down Timer" on page 21-10.

Use the **ip rip force-holddowntimer** command to configure the interval during which a RIP route remains in a forced hold-down state. Enter the command and the forced hold-down interval value, in seconds. For example, to set a forced hold-down interval value of 10 seconds you would enter:

```
-> ip rip force-holddowntimer 10
```

The valid forced hold-down timer range is **0** to **120**.

Use the **show ip rip** command to display the current forced hold-down timer value.

Configuring the RIP Update Interval

The RIP update interval defines the time interval, in seconds, when routing updates are sent out. This interval value must be less than or equal to one-third the value of the invalid timer.

Use the **ip rip update-interval** command to configure the interval during which a RIP route remains in an update state. Enter the command and the update interval value, in seconds. For example, to set an update interval value of 45 seconds, you would enter:

```
-> ip rip update-interval 45
```

The valid update interval range is **1** to **120**.

Configuring the RIP Invalid Timer

The RIP invalid timer value defines the time interval, in seconds, during which a route remains active in the Routing Information Base (RIB) before it is moved to the invalid state. This timer value must be at least three times the update interval value.

Use the **ip rip invalid-timer** command to configure the time interval that must elapse before an active route becomes invalid. Enter the command and the invalid timer value, in seconds. For example, to set an invalid interval value of 270 seconds you would enter:

```
-> ip rip invalid-timer 270
```

The invalid timer range is **3** to **360**.

Configuring the RIP Garbage Timer

The RIP garbage timer defines the time interval, in seconds, that must elapse before an expired route is removed from the RIB.

Note that during the garbage interval, the router advertises the route with a metric of INFINITY.

Use the **ip rip garbage-timer** command to configure the time interval after which an expired route is removed from the RIB. Enter the command and the garbage timer value, in seconds. For example, to set a garbage timer value of 180 seconds you would enter:

```
-> ip rip garbage-timer 180
```

The garbage timer range is **0** to **180**.

Configuring the RIP Hold-Down Timer

The RIP hold-down timer defines the time interval, in seconds, during which a route remains in the holddown state.

Whenever RIP detects a route with a higher metric than the route in the RIB, the route with the higher metric goes into the hold-down state. The route updates with a metric of INFINITY are excluded.

Use the **ip rip holddown-timer** command to configure the interval during which a RIP route remains in the hold-down state. Enter the command and the hold-down timer value, in seconds. For example, to set a hold-down timer value of 10 seconds you would enter:

```
-> ip rip holddown-timer 10
```

The hold-down timer range is **0** to **120**.

Reducing the Frequency of RIP Routing Updates

To optimize system performance, you can reduce the frequency of the RIP routing updates by increasing the length of the update, invalid, and garbage timers by about 50% above their default values. For example:

```
-> ip rip update-interval 45
-> ip rip invalid-timer 270
-> ip rip garbage-timer 180
```

Enabling a RIP Host Route

A host route differs from a network route, which is a route to a specific network. This command allows a direct connection to the host without using the RIP table. If a switch is directly attached to a host on a network, use the **ip rip host-route** command to enable a default route to the host. For example:

```
-> ip rip host-route
```

The default is to enable a default host route.

Use the **no ip rip host-route** command to disable the host route. Use the **show ip rip** command to display the current host route status.

Configuring Redistribution

It is possible to configure the RIP protocol to advertise routes learned from other routing protocols into the RIP network. Such a process is referred to as route redistribution and is configured using the **ip redist** command.

Redistribution uses route maps to control how external routes are learned and distributed. A route map consists of one or more user-defined statements that can determine which routes are allowed or denied access to the RIP network. In addition a route map can also contain statements that modify route parameters before they are redistributed.

When a route map is created, it is given a name to identify the group of statements that it represents. This name is required by the **ip redist** command. Therefore, configuring route redistribution involves the following steps:

1 Create a route map, as described in "Using Route Maps" on page 21-12.

2 Configure redistribution to apply a route map, as described in "Configuring Route Map Redistribution" on page 21-15.

Using Route Maps

A route map specifies the criteria that are used to control redistribution of routes between protocols. Such criteria is defined by configuring route map statements. There are three different types of statements:

- **Action**. An action statement configures the route map name, sequence number, and whether or not redistribution is permitted or denied based on route map criteria.
- **Match.** A match statement specifies criteria that a route must match. When a match occurs, then the action statement is applied to the route.
- **Set.** A set statement is used to modify route information before the route is redistributed into the receiving protocol. This statement is only applied if all the criteria of the route map is met and the action permits redistribution.

The **ip route-map** command is used to configure route map statements and provides the following **action**, **match**, and **set** parameters:

ip route-map action ...	ip route-map match ...	ip route-map set ...
permit **deny**	**ip-address** **ip-nexthop** **ipv6-address** **ipv6-nexthop** **tag** **ipv4-interface** **ipv6-interface** **metric** **route-type**	**metric** **metric-type** **tag** **community** **local-preference** **level** **ip-nexthop** **ipv6-nexthop**

Refer to the "IP Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the **ip route-map** command parameters and usage guidelines.

Once a route map is created, it is then applied using the **ip redist** command. See "Configuring Route Map Redistribution" on page 21-15 for more information.

Creating a Route Map

When a route map is created, it is given a name (up to 20 characters), a sequence number, and an action (permit or deny). Specifying a sequence number is optional. If a value is not configured, then the number 50 is used by default.

To create a route map, use the **ip route-map** command with the **action** parameter. For example,

```
-> ip route-map ospf-to-rip sequence-number 10 action permit
```

The above command creates the ospf-to-rip route map, assigns a **sequence number** of 10 to the route map, and specifies a **permit** action.

To optionally filter routes before redistribution, use the **ip route-map** command with a **match** parameter to configure match criteria for incoming routes. For example,

```
-> ip route-map ospf-to-rip sequence-number 10 match tag 8
```

The above command configures a match statement for the ospf-to-rip route map to filter routes based on their tag value. When this route map is applied, only OSPF routes with a tag value of eight are redistributed into the RIP network. All other routes with a different tag value are dropped.

Note. Configuring match statements is not required. However, if a route map does not contain any match statements and the route map is applied using the **ip redist** command, the router redistributes *all* routes into the network of the receiving protocol.

To modify route information before it is redistributed, use the **ip route-map** command with a **set** parameter. For example,

```
-> ip route-map ospf-to-rip sequence-number 10 set tag 5
```

The above command configures a set statement for the ospf-to-rip route map that changes the route tag value to five. Because this statement is part of the ospf-to-rip route map, it is only applied to routes that have an existing tag value equal to eight.

The following is a summary of the commands used in the above examples:

```
-> ip route-map ospf-to-rip sequence-number 10 action permit
-> ip route-map ospf-to-rip sequence-number 10 match tag 8
-> ip route-map ospf-to-rip sequence-number 10 set tag 5
```

To verify a route map configuration, use the **show ip route-map** command:

```
-> show ip route-map
Route Maps: configured: 1 max: 200
Route Map: ospf-to-rip Sequence Number: 10 Action permit
  match tag 8
  set tag 5
```

Deleting a Route Map

Use the **no** form of the **ip route-map** command to delete an entire route map, a route map sequence, or a specific statement within a sequence.

To delete an entire route map, enter **no ip route-map** followed by the route map name. For example, the following command deletes the entire route map named redistipv4:

```
-> no ip route-map redistipv4
```

To delete a specific sequence number within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the actual number. For example, the following command deletes sequence 10 from the redistipv4 route map:

```
-> no ip route-map redistipv4 sequence-number 10
```

Note that in the above example, the redistripv4 route map is not deleted. Only those statements associated with sequence 10 are removed from the route map.

To delete a specific statement within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the sequence number for the statement, then either **match** or **set** and the match or set parameter and value. For example, the following command deletes only the match tag 8 statement from route map redistipv4 sequence 10:

```
-> no ip route-map redistipv4 sequence-number 10 match tag 8
```

Configuring Route Map Sequences

A route map consists of one or more sequences of statements. The sequence number determines which statements belong to which sequence and the order in which sequences for the same route map are processed.

To add match and set statements to an existing route map sequence, specify the same route map name and sequence number for each statement. For example, the following series of commands creates route map rm_1 and configures match and set statements for the rm_1 sequence 10:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 set metric 1
```

To configure a new sequence of statements for an existing route map, specify the same route map name but use a different sequence number. For example, the following command creates a new sequence 20 for the rm_1 route map:

```
-> ip route-map rm_1 sequence-number 20 action permit
-> ip route-map rm_1 sequence-number 20 match ipv4-interface to-finance
-> ip route-map rm_1 sequence-number 20 set metric 5
```

The resulting route map appears as follows:

```
-> show ip route-map rm_1
Route Map: rm_1 Sequence Number: 10 Action permit
  match tag 8
  set metric 1
Route Map: rm_1 Sequence Number: 20 Action permit
  match ipv4 interface to-finance
  set metric 5
```

Sequence 10 and sequence 20 are both linked to route map rm_1 and are processed in ascending order according to their sequence number value. Note that there is an implied logical OR between sequences. As a result, if there is no match for the tag value in sequence 10, then the match interface statement in sequence 20 is processed. However, if a route matches the tag 8 value, then sequence 20 is not used. The set statement for whichever sequence was matched is applied.

A route map sequence contains multiple match statements. If these statements are of the same kind (e.g., match tag 5, match tag 8, etc.) then a logical OR is implied between each like statement. If the match statements specify different types of matches (e.g. match tag 5, match ip4 interface to-finance, etc.), then a logical AND is implied between each statement. For example, the following route map sequence redistributes a route if its tag is either 8 or 5:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
```

The following route map sequence redistributes a route if the route has a tag of 8 or 5 *and* the route was learned on the IPv4 interface to-finance:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 match ipv4-interface to-finance
```

Configuring Access Lists

An IP access list provides a convenient way to add multiple IPv4 or IPv6 addresses to a route map. Using an access list avoids having to enter a separate route map statement for each individual IP address. Instead, a single statement is used that specifies the access list name. The route map is then applied to all the addresses contained within the access list.

Configuring an IP access list involves two steps: creating the access list and adding IP addresses to the list. To create an IP access list, use the **ip access-list** command (IPv4) or the **ipv6 access-list** command (IPv6) and specify a name to associate with the list. For example,

```
-> ip access-list ipaddr
-> ipv6 access-list ip6addr
```

To add addresses to an access list, use the **ip access-list address** (IPv4) or the **ipv6 access-list address** (IPv6) command. For example, the following commands add addresses to an existing access list:

```
-> ip access-list ipaddr address 16.24.2.1/16
-> ipv6 access-list ip6addr address 2001::1/64
```

Use the same access list name each time the above commands are used to add additional addresses to the same access list. In addition, both commands provide the ability to configure if an address and/or its matching subnet routes are permitted (the default) or denied redistribution. For example:

```
-> ip access-list ipaddr address 16.24.2.1/16 action deny redist-control all-
subnets
-> ipv6 access-list ip6addr address 2001::1/64 action permit redist-control no-
subnets
```

For more information about configuring access list commands, see the “IP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Route Map Redistribution

The **ip redist** command is used to configure the redistribution of routes from a source protocol into the RIP destination protocol. This command is used on the RIP router that performs the redistribution.

A source protocol is a protocol from which the routes are learned. A destination protocol is the one into which the routes are redistributed. Make sure that both protocols are loaded and enabled before configuring redistribution.

Redistribution applies criteria specified in a route map to routes received from the source protocol. Therefore, configuring redistribution requires an existing route map. For example, the following command configures the redistribution of OSPF routes into the RIP network using the ospf-to-rip route map:

```
-> ip redist ospf into rip route-map ospf-to-rip
```

RIP routes received by the router interface are processed based on the contents of the ospf-to-rip route map. Routes that match criteria specified in this route map are either allowed or denied redistribution into the RIP network. The route map can also specify the modification of route information before the route is redistributed. See "Using Route Maps" on page 21-12 for more information.

To remove a route map redistribution configuration, use the **no** form of the **ip redist** command. For example:

```
-> no ipv6 redist ospf into rip route-map ospf-to-rip
```

Use the **show ip redist** command to verify the redistribution configuration:

```
-> show ip redist

Source       Destination
Protocol     Protocol     Status    Route Map
------------+------------+---------+--------------------
LOCAL4       RIP          Enabled   rip_1
LOCAL4       OSPF         Enabled   ospf_2
LOCAL4       BGP          Enabled   bgp_3
RIP          OSPF         Enabled   ospf-to-rip
```

Configuring the Administrative Status of the Route Map Redistribution

The administrative status of a route map redistribution configuration is enabled by default. To change the administrative status, use the **status** parameter with the **ip redist** command. For example, the following command disables the redistribution administrative status for the specified route map:

```
-> ip redist ospf into rip route-map ospf-to-rip admin-state disable
```

The following command example enables the administrative status:

```
-> ip redist ospf into rip route-map ospf-to-rip admin-state enable
```

Route Map Redistribution Example

The following example configures the redistribution of OSPF routes into a RIP network using a route map (ospf-to-rip) to filter specific routes:

```
-> ip route-map ospf-to-rip sequence-number 10 action deny
-> ip route-map ospf-to-rip sequence-number 10 match tag 5
-> ip route-map ospf-to-rip sequence-number 10 match route-type external type2

-> ip route-map ospf-to-rip sequence-number 20 action permit
-> ip route-map ospf-to-rip sequence-number 20 match ipv4-interface intf_ospf
-> ip route-map ospf-to-rip sequence-number 20 set metric 255

-> ip route-map ospf-to-rip sequence-number 30 action permit
-> ip route-map ospf-to-rip sequence-number 30 set tag 8

-> ipv6 redist ospf into rip route-map ospf-to-rip
```

The resulting ospf-to-rip route map redistribution configuration does the following:

- Denies the redistribution of Type 2 external OSPF routes with a tag set to five.
- Redistributes into RIP all routes learned on the intf_ospf interface and sets the metric for such routes to 255.
- Redistributes into RIP all other routes (those not processed by sequence 10 or 20) and sets the tag for such routes to eight.

RIP Security

By default, there is no authentication used for a RIP. However, you can configure a password for a RIP interface. To configure a password, you must first select the authentication type (simple or MD5), and then configure a password.

Configuring Authentication Type

If simple or MD5 password authentication is used, both switches on either end of a link must share the same password. Use the **ip rip interface auth-type** command to configure the authentication type. Enter the name of the RIP interface, and then enter an authentication type:

- **none.** No authentication is used.
- **simple.** Simple password authentication is used.
- **md5.** MD5 authentication is used.

For example, to configure the RIP interface rip-1 for simple authentication you would enter:

```
-> ip rip interface rip-1 auth-type simple
```

To configure the RIP interface rip-1 for MD5 authentication you would enter:

```
-> ip rip interface rip-1 md5 auth-type md5
```

Configuring Passwords

If you configure simple or MD5 authentication you must configure a text string that is used as the password for the RIP interface. If a password is used, all switches that are intended to communicate with each other must share the same password.

After configuring the interface for simple authentication as described above, configure the password for the interface by using the **ip rip interface auth-key** command. Enter the IP address of the RIP interface, and then enter a 16-byte text string. For example to configure a password “nms” you would enter:

```
-> ip rip interface rip-1 auth-key nms
```

Verifying the RIP Configuration

A summary of the show commands used for verifying the RIP configuration is given here:

show ip rip	Displays the RIP status and general configuration parameters (e.g., forced hold-down timer).
show ip rip routes	Displays the RIP routing database. The routing database contains all the routes learned through RIP.
show ip rip interface	Displays the RIP interface status and configuration.
show ip rip peer	Displays active RIP neighbors (peers).
show ip redist	Displays the currently configured RIP redistribution filters.

For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

22 Configuring BFD

An increasingly important requirement of networking equipment is to rapidly detect communication failures between network systems to quickly establish alternative paths and reduce network convergence time. Data link hardware such as SONET alarms make failure detection fairly easy and quick. However, some media, such as Ethernet, do not support such kind of signaling, and some media can not detect certain kinds of failures in the path, such as failing interfaces or forwarding engine components.

In the absence of such signaling hardware, networks resort to using simple “Hello” mechanisms to detect failures in the communication pathways between adjacent systems. One such mechanism is the Bidirectional Forwarding Detection (BFD) protocol.

BFD protocol is a fairly simple and quick Hello protocol; it can be configured on the interfaces with routing protocols to rapidly detect faults in the bidirectional paths between adjacent forwarding engines, including data link(s) and forwarding engines. BFD is not intended to directly control liveliness information; instead, the application provides parameters and BFD supplies the state of the session. It acts in an advisory role to the control protocols. It provides a low overhead alternative to detect faults for all media types and routing protocols in a variety of network environments and topologies. BFD protocol sessions can be initiated for any remote IP address reachable through outgoing IP interface ports.

In This Chapter

This chapter describes the basic components of BFD and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration information and procedures described in this chapter include:

- Global Configuration (see page 22-14).
- Interface Level Configuration (see page 22-14).
- BGP Level Configuration (see page 22-18).
- IS-IS Level Configuration (see page 22-21).
- OSPF Level Configuration (see page 22-23).
- PIM Level Configuration (see page 22-26).
- VRRP Level Configuration (see page 22-30).
- Static Routing Level Configuration (see page 22-31).

BFD Defaults

The following table shows the default settings for the configurable global BFD parameters.

Parameter Description	Command	Default Value/Comments
BFD global status for the switch	**ip bfd admin-state**	Disabled
Global transmit time interval for BFD control packets	**ip bfd transmit**	300 milliseconds
Global receive time interval for BFD control packets.	**ip bfd receive**	300 milliseconds
Global BFD detection time multiplier	**ip bfd multiplier**	3
Global BFD echo packet time interval	**ip bfd echo-interval**	300 milliseconds

The following table shows the default settings for the configurable BFD interface parameters.

Parameter Description	Command	Default Value/Comments
Administrative status of an IPv4 or IPv6 BFD interface	**ip|ipv6 bfd interface admin-state**	Disabled
Transmit time interval for an IPv4 or IPv6 BFD interface	**ip|ipv6 bfd interface transmit**	300 milliseconds
Receive time interval for an IPv4 or IPv6 BFD interface	**ip|ipv6 bfd interface receive**	300 milliseconds
Detection time multiplier for an IPv4 or IPv6 BFD interface	**ip|ipv6 bfd interface multiplier**	3
Echo time interval for an IPv4 or IPv6 BFD interface	**ip|ipv6 bfd interface echo-interval**	300 milliseconds

The following table shows the default settings for the configurable BFD parameters at the protocol level.

Parameter Description	Command	Default Value/Comments
BFD status for the BGP protocol	**bgp neighbor check-first-as**	Disabled
BFD session status with all BGP neighbors	**ip bgp bfd-state all-neighbors**	Disabled
BFD session status for a specific IPv4 or IPv6 BGP neighbor	**ip|ipv6 bgp neighbor bfd-state**	Disabled
BFD status for the IS-IS protocol	**ip isis bfd-state**	Disabled
BFD session status with all IS-IS VLANs	**ip isis bfd-state all-vlans**	Disabled
BFD session status for a specific IS-IS VLAN	**ip isis vlan bfd-state**	Disabled
BFD status for the OSPF or OSPFv3 protocol	**ip ospf bfd-state** **ipv6 ospf bfd-state**	Disabled
BFD status for an OSPF or OSPFv3 interface	**ip ospf interface bfd-state** **ipv6 ospf interface bfd-state**	Disabled

Parameter Description	Command	Default Value/Comments
BFD session status with all OSPF or OSPFv3 neighbors of the corresponding interface which are greater than or equal to “2-way” state	**ip ospf interface bfd-state all-neighbors** **ipv6 ospf interface bfd-state all-neighbors**	Enabled
BFD status for the IPv4 or IPv6 PIM protocol	**ip pim bfd-state** **ipv6 pim bfd-state**	Disabled
BFD status for an IPv4 or IPv6 PIM interface.	**ip pim interface bfd-state** **ipv6 pim interface bfd-state**	Disabled
BFD status for an IPv4 or IPv6 static route.	**ip static-route bfd-state** **ipv6 static-route bfd-state**	Enabled
BFD status for VRRP protocol	**vrrp bfd-state**	Disabled
BFD status for a VRRP tracking policy.	**vrrp track**	Enabled

Quick Steps for Configuring BFD

Configuring BFD involves:

- *Optional*: Configuring BFD explicitly on the IP interfaces.
- Configuring Layer 3 protocols to use BFD (see "Quick Steps for Configuring BFD Support for Layer 3 Protocols" on page 22-6).

Note. Configuring a BFD session explicitly with an IP interface name is optional, and must be used if user-defined BFD session parameters need to be applied. All the steps for explicit configuration are mentioned as optional.

If BFD is not explicitly configured, the default BFD global session parameters (transmit, receive and echo intervals) are applied to the BFD sessions.

The following steps provide a brief tutorial for configuring a BFD session and related parameters:

1 Configure a BFD session on an IPv4 or IPv6 interface using the **ip|ipv6 bfd interface** command. For example:

```
-> ip bfd interface bfd_int_1
-> ipv6 bfd interface bfd_int_2
```

2 *Optional*: Configure a global transmit time interval for all BFD sessions using the **ip bfd transmit** command. This command defines a default transmit value that is automatically applied when a BFD session is created. For example:

```
-> ip bfd transmit 500
```

3 *Optional*: Configure the transmit time interval for a specific BFD session using the **ip|ipv6 bfd interface transmit** command. The value set with this command overrides the global transmit value configured for the routing instance. For example:

```
-> ip bfd interface bfd-vlan-101 transmit 500
-> ipv6 bfd interface bfd-vlan-201 transmit 500
```

4 *Optional*: Configure a global receive time interval for all BFD sessions using the **ip bfd receive** command. This command defines a default receive time value that is automatically applied when a BFD session is created. For example:

```
-> ip bfd receive 500
```

5 *Optional*: Configure the receive time interval for a specific BFD session using the **ip|ipv6 bfd interface receive** command. The value set with this command overrides the global receive time value configured for the routing instance:

```
-> ip bfd interface bfd-vlan-101 receive 500
-> ipv6 bfd interface bfd-vlan-201 receive 500
```

6 *Optional*: Configure a global detection time multiplier value for all BFD sessions using the **ip bfd multiplier** command. For example:

```
-> ip bfd multiplier 5
```

7 *Optional*: Configure the session detection time multiplier value for a specific BFD session using the **ip|ipv6 bfd interface multiplier** command. For example:

```
-> ip bfd interface bfd-vlan-101 multiplier 5
-> ipv6 bfd interface bfd-vlan-201 multiplier 5
```

8 *Optional*: Configure the global BFD echo packet time interval using the **ip bfd echo-interval** command. This command defines a default echo packet time value that is automatically applied when a BFD session is created. For example:

```
-> ip bfd echo-interval 500
```

9 *Optional*: Configure the echo time interval for a specific BFD session using the **ip|ipv6 bfd interface echo-interval** command. The echo time interval value set with this command overrides the global echo time interval configured for the routing instance. For example:

```
-> ip bfd interface bfd-vlan-101 echo-interval 500
-> ipv6 bfd interface bfd-vlan-201 echo-interval 500
```

10 *Optional*: Enable the administrative status of a BFD interface using the **ip|ipv6 bfd interface admin-state** command. For example:

```
-> ip bfd interface bfd-vlan-101 admin-state enable
-> ipv6 bfd interface bfd-vlan-201 admin-state enable
```

Note. BFD parameters are not configurable once the BFD administrative status is enabled on the interface.

---

11 Enable the BFD protocol for the routing instance globally using the **ip bfd admin-state** command. For example:

```
-> ip bfd admin-state enable
```

Note. *Optional*. To verify the global BFD configuration for the switch, use the **show ip bfd** command. For example:

```
-> show ip bfd
BFD Version Number              = 1,
Admin Status                    = Enabled,
Desired Transmit Interval        = 300,
Minimum Receive Interval        = 300,
Detection Time Multiplier       = 3,
Minimum Echo Receive Interval   = 300,
Applications Registered         = STATIC-ROUTING OSPF
```

Verify the BFD interface session status and configuration using the **show ip|ipv6 bfd interfaces** command. For example:

```
-> show ip bfd interfaces bfd-intf1
Interface Name                   = bfd-intf1,
Interface IP Address             = 100.1.1.1,
Admin Status                     = Enabled,
Desired Transmit Interval        = 300,
Minimum Receive Interval         = 300,
Detection Time Multiplier        = 3,
Minimum Echo Receive Interval    = 300,
Authentication Present           = No,
Oper Status                      = UP
```

```
-> show ipv6 bfd interfaces bfd-intf3
Interface Name                    = bfd-intf3
Interface IP Address              = fe80::2efa:a2ff:fe13:e402,
Admin Status                      = Disabled,
Desired Transmit Interval         = 300,
Minimum Receive Interval          = 300,
Detection Time Multiplier         = 3,
Minimum Echo Receive Interval     = 300,
Authentication Present            = No,
Oper Status                       = DOWN
```

See the "BFD Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in this display.

Quick Steps for Configuring BFD Support for Layer 3 Protocols

BFD runs on top of Layer 3 protocol traffic that is forwarded between two systems. This implementation of BFD supports the following protocols:

- IPv4 and IPv6 BGP
- IPv4 and IPv6 IS-IS
- OSPF and OSPFv3
- IPv4 and IPv6 PIM
- IPv4 and IPv6 Static routes
- VRRP Tracking of IPv4 and IPv6 interfaces.

Once the BFD configuration is in place (see "Quick Steps for Configuring BFD" on page 22-4), the steps described in the following sections are used to configure BFD interaction with the supported Layer 3 protocols.

Configuring BFD Support for BGP

1 Register BGP with the BFD protocol using the **bgp neighbor check-first-as** command. For example:

```
-> ip bgp bfd-state enable
```

2 Enable BFD for a specific IPv4 or IPv6 BGP neighbor using the **ip|ipv6 bgp neighbor bfd-state** command or for all BGP neighbors using the **ip bgp bfd-state all-neighbors** command. For example:

```
-> ip bgp neighbor 135.10.10.2 bfd-state enable
-> ipv6 bgp neighbor fe80::2efa:a2ff:fe13:e402 bfd-state enable
-> ip bgp bfd-state all-neighbors enable
```

Configuring BFD Support for IS-IS

1 Register IS-IS with the BFD protocol using the **ip isis bfd-state** command. For example:

```
-> ip isis bfd-state enable
```

2 Enable BFD for a specific IS-IS VLAN using the **ip isis vlan bfd-state** command or for all IS-IS VLANs using the **ip isis bfd-state all-vlans** command. For example:

```
-> ip isis vlan 10 bfd-state enable
-> ip isis bfd-state all-vlans enable
```

Configuring BFD Support for OSPF

1 Register OSPF with the BFD protocol using the **ip ospf bfd-state** command. For example:

```
-> ip ospf bfd-state enable
```

2 Enable BFD session on a specific OSPF interface using the **ip ospf interface bfd-state** command or on all OSPF interfaces using the **ip ospf bfd-state all-interfaces** command. For example:

```
-> ip ospf interface int1 bfd-state enable
-> ip ospf bfd-state all-interfaces
```

3 Establish BFD sessions with all OSPF DR neighbors in full states only or with all neighbors greater than or equal to the "2-way" state using the **ip ospf interface bfd-state drs-only** command or the **ip ospf interface bfd-state all-neighbors** command. For example:

```
-> ip ospf interface int1 bfd-state drs-only
-> ip ospf interface int1 bfd-state all-neighbors enable
```

Configuring BFD Support for OSPFv3

1 Register OSPFv3 with the BFD protocol using the **ipv6 ospf bfd-state** command. For example:

```
-> ipv6 ospf bfd-state enable
```

2 Enable BFD session on a specific OSPFv3 interface using the **ipv6 ospf interface bfd-state** command or on all OSPFv3 interfaces using the **ipv6 ospf bfd-state all-interfaces** command. For example:

```
-> ipv6 ospf interface int2 bfd-state enable
-> ipv6 ospf bfd-state all-interfaces
```

3 Establish BFD sessions with all OSPFv3 DR neighbors in full states only or with all neighbors greater than or equal to the "2-way" state using the **ipv6 ospf interface bfd-state drs-only** command or the **ipv6 ospf interface bfd-state all-neighbors** command. For example:

```
-> ipv6 ospf interface int2 bfd-state drs-only
-> ipv6 ospf interface int2 bfd-state all-neighbors enable
```

Configuring BFD Support for IPv4 PIM

1 Register IPv4 PIM with the BFD protocol using the **ip pim bfd-state** command. For example:

```
-> ip pim bfd-state enable
```

2 Enable BFD for a specific IPv4 PIM interface using the **ip pim interface bfd-state** command or for all IPv4 PIM interfaces using the **ip pim bfd-state all-interfaces** command. For example:

```
-> ip pim interface pimInt1 bfd-state enable
-> ip pim bfd-state all-interfaces enable
```

Configuring BFD Support for IPv6 PIM

1 Register IPv6 PIM with the BFD protocol using the **ipv6 pim bfd-state** command. For example:

```
-> ipv6 pim bfd-state enable
```

2 Enable BFD for a specific IPv6 PIM interface using the **ipv6 pim interface bfd-state** command or for all IPv6 PIM interfaces using the **ipv6 pim bfd-state all-interfaces** command. For example:

```
-> ipv6 pim interface pimInt1 bfd-state enable
-> ipv6 pim bfd-state all-interfaces enable
```

Configuring BFD Support for IPv4 Static Routes

Enable BFD support for a specific IPv4 static route using the **ip static-route bfd-state** command or for all IPv4 static routes using the **ip static-route all bfd-state** command. For example:

```
-> ip static-route 192.100.1.0/24 gateway 100.1.1.10 bfd-state enable
-> ip static-route all bfd-state enable
```

To create a BFD session for an IPv4 static route, make sure that:

- the gateway address does not match any of the local interface addresses on the switch
- BFD is enabled for the interface on which the gateway address exists.
- If multiple routes are configured with the same gateway address, only one BFD session is run.

Note. To display the IPv4 static routes on which BFD is enabled use the **show ip router database** command along with the **protocol static** option as shown below:

```
-> ip static-route 100.0.0.0/8 gateway 100.1.1.10 bfd-state enable

-> show ip router database protocol static
Legend: + indicates routes in-use
        b indicates BFD-enabled static route
        i indicates interface static route
        r indicates recursive static route, with following address in brackets

Destination         Gateway         Interface    Protocol  Metric  Tag   Misc-Info
-------------------+---------------+------------+--------+-------+-----+-----------
+b 100.0.0.0/8        100.1.1.10      v1001        STATIC    1     0
+  128.251.40.0/24    172.28.4.254    EMP          STATIC    1     0

Inactive Static Routes
   Destination       Gateway           Metric
-------------------+-----------------+---------
```

See the “IP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in this display.

Configuring BFD Support for IPv6 Static Routes

Enable BFD support for a specific IPv6 static route using the **ipv6 static-route bfd-state** command or for all IPv6 static routes using the **ipv6 static-route all bfd-state** command. For example:

```
-> ipv6 static-route 195:35::/64 gateway fe80::2d0:95ff:fe12:f470 bfd-state
enable
-> ipv6 static-route all bfd-state enable
```

To create a BFD session for an IPv6 static route, make sure that:

- the gateway address does not match any of the local interface addresses on the switch

- BFD is enabled for the interface on which the gateway address exists.
- If multiple routes are configured with the same gateway address, only one BFD session is run.

Note. To display the IPv6 static routes on which BFD is enabled use the **show ipv6 router database** command along with the **protocol static** option as shown below:

```
-> ipv6 static-route 2002::/16 gateway 2002:d423:2323::35 bfd-state enable

-> show ipv6 router database protocol static
Legend: + indicates routes in-use
        b indicates BFD-enabled static route

Destination/Prefix Gateway Address          Interface       Metric    Tag   Misc-Info
------------------+-----------------------+---------------+--------+-----+---------
+b 2002::/16         2002:d423:2323::35      v6if-6to4-137    1        0

Inactive Static Routes:
Vlan  Destination/Prefix Gateway Address          Metric    Tag   Misc-Info
-----+------------------+-----------------------+--------+-----+----------
  -    3333::/24          4444::                    1          0
```

See the “IPv6 Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in this display.

Configuring BFD Support for VRRP Track Policies

1. Register VRRP with the BFD protocol using the **vrrp bfd-state** command. For example:

```
-> vrrp bfd-state enable
```

2. Enable BFD for a specific track policy using the **vrrp track** command. For example:

```
-> vrrp track 2 address 10.1.1.1 bfd-state enable
-> vrrp track 5 address 213:100:1::56 bfd-state enable
```

Make sure that the track policy is associated with at least one of the virtual routers. In addition, note that the value of the address parameter should be a remote interface address. BFD cannot be configured for a local interface address.

Note. To display the VRRP tracking policies on which BFD is enabled, use the **show vrrp track** command.

```
-> show vrrp track

Track                              Admin       Oper            BFD
 ID          Policy                State       State     Pri   Status
-----+-----------------------+-------------+--------+-------+--------------+
  1           10.1.1.1        Enabled        Down       50    Enabled
  2           213:100:1::56    Enabled        Down       25    Enabled
```

See the “VRRP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in this display.

BFD Overview

Detecting communication failures as soon as possible is the first step in any network recovery process; until a failure is detected, network convergence can't begin. By rapidly detecting failures, BFD enables faster convergence of routing protocols particularly on shared media such as Ethernet.

The BFD protocol is very similar to the widely-used Hello mechanisms prevalent in a majority of routing protocols, with the exception that BFD tests bidirectional communication links, has smaller packets, and is focused exclusively on path-failure detection. BFD can also be less CPU-intensive in routers with distributed architecture because unlike routing protocol Hello packets, BFD packets can be processed on the interface modules rather than the control plane.

BFD protocol is a fairly simple Hello protocol designed to provide fast forwarding path failure detection that can be enabled at the interface and routing protocol levels. It helps in the verification of forwarding plane-to-forwarding plane connectivity (including links, interfaces, tunnels). It allows semantic separation of forwarding plane connectivity and control plane connectivity. BFD is a single mechanism that works independently of underlying media, data, and network protocols. It can be associated with any routing protocol running between two systems. Moreover, it requires no changes to the existing protocols.

This implementation of BFD can be associated with tracking of next hops with the BGP, OSPF, VRRP and other static route protocols.

Benefits of Using BFD For Failure Detection

It is more advantageous to implement BFD rather than reduce timer mechanisms for routing protocols due to the following reasons:

- BFD can detect failures in milliseconds without having to fine-tune routing protocol Hello timers.
- BFD is not tied to any particular routing protocol. As a result, BFD provides a generic and consistent failure detection mechanism for OSPF, BGP, VRRP Remote Tracking, and static routes.
- BFD is less CPU-intensive than reduced timer mechanisms for routing protocols.

How the BFD Protocol Works

A BFD session must be explicitly configured between two adjacent systems. Once BFD has been enabled on the interfaces and at the appropriate Layer 3 routing protocol level, a BFD session is created for the adjacent systems and BFD timers are negotiated between these systems.

The BFD protocol does not have a neighbor discovery mechanism to detect neighboring systems; protocols that BFD services notify BFD of devices to which it needs to establish sessions. For example, an OSPF implementation can request BFD to establish a session with a neighbor discovered using the OSPF Hello protocol.

Once a session is established, BFD peers—neighboring systems sharing a BFD interface—begin sending BFD control packets to each other over the bidirectional forwarding path. The packets are transmitted periodically at the negotiated rate. The BFD control packets function in a similar manner to that of an IGP Hello protocol, except at a more accelerated rate.

Each time a BFD control packet is successfully received through a BFD interface, the detect-timer for that session is reset to zero. As long as the BFD peer systems receive the control packets from each other within the negotiated time interval [(Detect Time Multiplier) * (Required Minimum Rx Interval)], the BFD session remains up. Any routing protocol associated with BFD maintains its adjacencies. BFD continues its periodic transmission of control packets at the negotiated rate.

In case a system stops receiving the packets within the predetermined time frame, some component in the bidirectional path to that particular system is assumed to have failed, and the BFD system simply informs its client protocol that a failure has occurred. It does this by sending rapid failure detection notices to respective registered routing protocols in the local router to initiate the router table recalculation process in order to accelerate routing convergence and network uptime.

In order to agree with its peers about how rapidly failure detection takes place, each system estimates the rate at which it can send and receive BFD control packets. This design also enables fast systems on shared medium with a slow system to detect failures more rapidly between fast systems while allowing the slow system to participate to the best of its ability.

Operational Mode and Echo Function

The BFD protocol offers two different modes of operation:

- Asynchronous mode
- Demand mode (not supported)

This implementation of BFD supports an Asynchronous control packet mode or an Asynchronous Echo function.

- When the Asynchronous control packet mode is activated, BFD neighbors periodically send BFD control packets to each other. A time interval for transmitting and receiving such packets is negotiated between the two BFD systems. If a neighboring system fails to receive a number of control packets continuously over a specific period of time, the session is considered down and BFD informs the appropriate routing protocol.
- The Asynchronous Echo function is used to verify the forwarding path between neighboring BFD systems. When active, a BFD system transmits Echo packets only (no control packets are sent) to a BFD neighbor, which then sends the packets back to the originating system along the forwarding path. If no Echo packets are received back from the BFD neighbor within a configured Echo time interval, the session is considered down.

VRRP and Static Routes only use the Asynchronous Echo function, so BFD sends only Echo packets. Other protocols (OSPF, IS-IS, BGP) use the Asynchronous control packet mode, so BFD initiates and maintains sessions by sending control packets.

Consider the following regarding how BFD operates between peers:

- Transmitting Echo packets is only allowed over a single hop; transmitting BFD control packets is allowed over multiple hops.
- The Echo function does not require a BFD session to run; instead, the function is activated when BFD is enabled for the switch that is going to send the Echo packets. The peer switches that are going to receive the Echo packets do not require a BFD configuration since this function is not a BFD session.

BFD Packet Formats

The detection packets BFD sends are UDP packets which are of two types: BFD control packets and Echo packets.

BFD Control Packets

There is no specific encapsulation type for BFD control packets; instead, the BFD IETF RFC-5880 recommends an encapsulation type that is “appropriate to the medium and network” used. This

implementation of BFD for IPv4 routing protocols (BGP, OSPF, VRRP Remote Tracking, and static routes), associates BFD control packets in UDP packets using destination port 3784 and a source port in the range of 49152 to 65535.

Note. The BFD control packet has a mandatory section and an optional authentication section. Authentication is not supported in this implementation of the BFD protocol.

BFD Echo Packets

There is no specific definition for Echo packet format. The only requirement is that the transmitting system is able to use the packet contents to distinguish between the various BFD sessions so that packets are correctly processed for the appropriate session.

This implementation of BFD associates Echo packets in UDP packets using port 3785 and the IP address of the transmitting interface. The contents of the Echo packet is defined as follows:

Field	Description
Version	The version number of the BFD protocol.
My Discriminator	An identifier for the BFD session connecting to the local side.
Sequence Number	The sequence number for this packet. This value is incremented for each successive packet transmitted for a session.

BFD Session Establishment

There are three states through which a BFD session normally proceeds: two for establishing a session (Up and Init state) and one for tearing down a session (Down state). In addition, an AdminDown state exists to administratively take down a session.

BFD uses a three-way handshake to establish sessions and guarantee that each BFD peer is aware of all the state changes. The transmitting system fills the state field in the transmitted BFD control packet with its current session state. To establish a session, the receiving peer system changes its session state based on the state field value in the received BFD control packet and its own session status.

A Down state means that a session is down or has been recently created. A session remains down until the remote system sends a packet with any state other than an up state. If a BFD packet with the state field set to down is received by the local system that is also in a down state, the session advances to Init state; if that packet signals Init state, the session advances to Up state.

Init signals that there is communication between the systems and that the local system wishes to start a session but the remote system has not yet acknowledged it. The session stays at Init until the local system receives a control packet with Init or Up in its state field (in which case the session state moves to Up) or until the detection time limit is reached.(in which case the remote system is then considered unreachable and the state moves to Down)

An Up state indicates that a BFD session has been created and both BFD peers are communicating with each other. The BFD session continues to remain in this state until connectivity fails and the state moves to Down or until the BFD session is taken down administratively.

Demultiplexing

Each BFD session must be able to uniquely identify itself and received BFD packets among the myriad of BFD sessions that are running. Each BFD peer must choose an identifying and unique discriminator value. This value is sent in the “My Discriminator” field of the BFD control packet, and is reflected back in the “Your Discriminator” field of the control packet sent from the remote peer. Once the system has echoed the respective “Your Discriminator” value back to its peer, the packets are demultiplexed (converted back into their original separate signals).

BFD Timer Negotiation

The BFD control packet contains information about how quickly a system would like to send packets to its peer, as well as how rapidly it is willing to receive packets from the peer. The BFD detection time is not carried explicitly in the protocol, but rather, it is determined by the receiving system independently based on the transmission interval (TX) and Detection Time Multiplier that have been negotiated.

The Detection Time Multiplier field value is approximately the number of packets that must be missed in order to declare a session down. In Asynchronous mode, detection times can be different in each direction. The local system detection time in this mode equals the value of Detection Time Multiplier received from the remote system multiplied by the negotiated transmission interval (TX). Because the time values for BFD control packet transmissions and session detection are being constantly negotiated by the participating BFD peers, they can be changed at any time. They are also independent in each direction for each session.

To change the rate at which BFD control packets are received, you can change the Required Min RX Interval at any time to any value. This new value is sent in the next outgoing packet so that the remote system can accommodate the changes made. Similarly, to change the rate at which BFD control packets are transmitted, you can change the Desired Min TX Interval at any time to any value.

With some exceptions, a system cannot transmit control packets with an interval shorter than the larger value of the TX interval and RX interval fields. This means that the system with the slower rate determines the BFD control packet transmission speed.

Configuring BFD

Configuring BFD for your network requires the following approach:

1 *Optional*: Configure a BFD session and related session parameter values. Once configured, enable all participating BFD sessions *before* configuring BFD interoperability with the supported Layer 3 protocols. See "Configuring BFD Session Parameters" on page 22-14 for more information.

2 Configure BFD support for the Layer 3 protocols for which BFD establishes sessions. This implementation of BFD supports the IPv4 and IPv6 versions of BGP, IS-IS, OSPF, PIM, VRRP remote tracking, and static routes. See "Configuring BFD Support for Layer 3 Protocols" on page 22-18 for more information.

At the end of the chapter is a simple BFD network diagram with instructions on how it can be created on a router-by-router basis. See "BFD Application Example" on page 22-33 for more information.

Configuring BFD Session Parameters

When a BFD session is created, default values are automatically set for these parameters. However, it is possible to change these parameter values globally or for a specific BFD session. The following BFD session parameter values are used to create, monitor, and negotiate BFD sessions between peers.

- BFD session status (see "Configuring a BFD Session" on page 22-14).
- Transmit time interval (see "Configuring the BFD Transmit Time interval" on page 22-15).
- Receive time interval (see "Configuring the BFD Receive Time Interval" on page 22-15).
- Multiplier (see "Configuring the BFD Multiplier" on page 22-16).
- Echo interval (see "Configuring the BFD Echo interval" on page 22-15).

Note. Once the default state of the BFD session is changed and the session is enabled, parameter values are no longer configurable. To subsequently change parameter values, disable the BFD session. See "Enabling or Disabling BFD Status" on page 22-16 for more information.

Configuring a BFD Session

To configure a BFD session, use the **ip|ipv6 bfd interface** command and specify an existing IPv4 or IPv6 interface name. For example:

```
-> ip bfd interface bfd-vlan-101
-> ipv6 bfd interface bfd-vlan-201
```

The above commands configure and IPv4 BFD interface with the name "bfd-vlan-101" and an IPv6 BFD interface with the name "bfd-vlan-201". See "Enabling or Disabling BFD Status" on page 22-16 for more information.

To delete the BFD session, use the **no** form of the above commands. For example:

```
-> no ip bfd interface bfd-vlan-101
-> no ipv6 bfd interface bfd-vlan-201
```

The above commands delete the BFD session on "bfd-vlan-101" and "bfd-vlan-201".

Note. The BFD interface session must be associated to an existing IPv4 or IPv6 interface that is configured with an IPv4 or IPv6 address.

Configuring the BFD Transmit Time interval

Transit Time Interval is the minimum amount of time that BFD waits between each successive transmission of control packets. BFD allows you to change the default value and set the transmit time interval from the valid range.

To change the global transmit time interval for BFD control packets, use the **ip bfd transmit** command. For example:

```
-> ip bfd transmit 500
```

The above command changes the global transmit time interval to 500 msecs.

To change the transmit time interval for a specific BFD interface session, use the **ip|ipv6 bfd interface transmit** command along with the name and transmit time interval in milliseconds. For example:

```
-> ip bfd interface bfd-vlan-101 transmit 500
-> ipv6 bfd interface bfd-vlan-201 transmit 500
```

The above command changes the transmit time interval value to 500 msecs on “bfd-vlan-101” and “bfd-vlan-201”.

The global transmit time interval serves as the default interval value for transmitting BFD control packets. This default value is overridden when a specific **transmit** value is configured.

Configuring the BFD Receive Time Interval

Receive Time Interval is the minimum amount of time that BFD waits to receive control packets before determining if there is a problem. BFD allows you to change the default value and set the receive time interval from the valid range.

To change the global receive time interval for BFD control packets, use the **ip bfd receive** command. For example:

```
-> ip bfd receive 500
```

The above command configures the global receive time interval of 500 msecs.

To change the receive time interval for BFD control packets, use the **ip|ipv6 bfd interface receive** command. For example:

```
-> ip bfd interface bfd-vlan-101 receive 500
-> ipv6 bfd interface bfd-vlan-201 receive 500
```

The above command changes the receive time interval value to 500 msecs on “bfd-vlan-101” and “bfd-vlan-201”.

The global receive time interval serves as the default interval value for receiving BFD control packets. The default interval value is overridden when a specific **receive** value is configured.

Configuring the BFD Echo interval

The time interval between received BFD echo packets is configurable and applies when the echo function is enabled. When this function is active, a stream of Echo packets is sent to a peer, which then loops these

back to the sender without processing them through its forwarding path. If the sender does not receive several continuous echo packets from its peer, the BFD session is declared down.

To change the default value of the global BFD echo packet time interval, use the **ip bfd echo-interval** command. For example:

```
-> ip bfd echo-interval 500
```

The above command sets the echo interval to 500 milliseconds globally on all BFD sessions.

To change the BFD echo time interval for a particular BFD session, use the **ip|ipv6 bfd interface echo-interval** command. For example:

```
-> ip bfd interface bfd-vlan-101 echo-interval 500
-> ipv6 bfd interface bfd-vlan-201 echo-interval 500
```

The above command configures the echo time interval value to 500 milliseconds on “bfd-vlan-101” and “bfd-vlan-201”.

The global echo packet time interval serves as the default interval value. The default interval value is overridden when a specific value is configured.

Configuring the BFD Multiplier

The BFD multiplier value is used to calculate the BFD detection time in asynchronous mode. The detection time between neighbors is calculated by multiplying the negotiated transmit time interval by the dead interval multiplier. When an interface stops receiving packets from a neighbor, the interface uses the detection time value to determine how long to wait before declaring that the BFD session is down.

The BFD multiplier parameter can be configured globally for all BFD configured interfaces as well as for a specific interface.

To set or change the default global detection time multiplier value for all BFD sessions, use the **ip bfd multiplier** command. For example:

```
-> ip bfd multiplier 5
```

The above command assigns a multiplier value of 5 to all BFD sessions.

To change the BFD multiplier for a specific session, use the **ip|ipv6 bfd interface multiplier** command. For example:

```
-> ip bfd interface bfd-vlan-101 multiplier 5
-> ipv6 bfd interface bfd-vlan-201 multiplier 5
```

The above command assigns a multiplier value of 5 to “bfd-vlan-101” and “bfd-vlan-201”.

Enabling or Disabling BFD Status

As BFD is globally disabled for the routing instance, to enable the global BFD status, use the **ip bfd admin-state** command. For example:

```
-> ip bfd admin-state enable
```

To disable the global BFD status for the routing instance, use the **ip bfd admin-state** command with the **disable** keyword. For example:

```
-> ip bfd admin-state disable
```

The above command disables BFD globally on the routing instance. Note that disabling BFD does not remove the existing BFD configuration from the routing instance. Also, when BFD is globally disabled, all BFD functionality is disabled for the routing instance, but configuring BFD is still allowed.

To enable a BFD session, use the **ip|ipv6 bfd interface admin-state** command. For example:

```
-> ip bfd interface bfd-vlan-101 admin-state enable
-> ipv6 bfd interface bfd-vlan-201 admin-state enable
```

The above command enables the administrative status of “bfd-vlan-101” and “bfd-vlan-201”.

Note that a BFD session must be disabled before any of its parameters can be changed. To disable a BFD session, use the **ip bfd interface admin-state** command or the **ipv6 bfd interface admin-state** command with the **disable** keyword. For example:

```
-> ip bfd interface bfd-vlan-101 admin-state disable
-> ipv6 bfd interface bfd-vlan-201 admin-state disable
```

To verify the global BFD status and configuration for the switch, use the **show ip bfd** command,. For example:

```
-> show ip bfd
BFD Version Number              = 1,
Admin Status                    = Enabled,
Desired Transmit Interval        = 300,
Minimum Receive Interval        = 300,
Detection Time Multiplier       = 3,
Minimum Echo Receive Interval   = 300,
Applications Registered         = STATIC-ROUTING OSPF
```

The above command shows that BFD is registered with the OSPF protocol and has a transmit interval of 300 msecs, receive interval of 300 msecs, multiplier 3, and echo interval of 300 msecs.

To verify the BFD status and configuration, use the **show ip|ipv6 bfd interfaces** command. For example:

```
-> show ip bfd interfaces

    Interface          Admin         Tx      Min Rx    Min EchoRx Detect  Oper
      Name             Status     Interval  Interval   Interval    Mult  Status
---------------------+----------+----------+----------+----------+------+------
bfd-intf1              enabled    300        300        300         3       UP
bfd-intf2              enabled    300        300        300         3       UP

-> show ipv6 bfd interfaces
    Interface          Admin         Tx      Min Rx    Min EchoRx Detect  Oper
      Name             Status     Interval  Interval   Interval    Mult  Status
---------------------+----------+----------+----------+----------+------+------
bfd-intf3             disabled    300        300        300        3      DOWN
```

The output above displays the interfaces participating in the BFD sessions, along with their IPv4 or IPv6 interface names and respective BFD session parameters. To see additional detail for a specific interface, use the **show ip|ipv6 bfd interfaces** command and specify an interface name. For example:

```
-> show ip bfd interfaces bfd-intf1
Interface Name                  = bfd-intf1,
Interface IP Address            = 100.1.1.1,
Admin Status                    = Enabled,
Desired Transmit Interval       = 300,
Minimum Receive Interval        = 300,
```

```
Detection Time Multiplier        = 3,
Minimum Echo Receive Interval    = 300,
Authentication Present           = No,
Oper Status                      = UP

-> show ipv6 bfd interfaces bfd-intf3
Interface Name                   = bfd-intf3
Interface IP Address             = fe80::2efa:a2ff:fe13:e402,
Admin Status                     = Disabled,
Desired Transmit Interval        = 300,
Minimum Receive Interval         = 300,
Detection Time Multiplier        = 3,
Minimum Echo Receive Interval    = 300,
Authentication Present           = No,
Oper Status                      = DOWN
```

Configuring BFD Support for Layer 3 Protocols

After a BFD session is configured on all interfaces or on a specific set of individual interfaces, the next step is to configure BFD interoperability with the supported Layer 3 protocols (BGP, IS-IS, OSPF, PIM, VRRP Tracking, Static Routes). BFD interoperability with Layer 3 protocols is configurable at the router level to enable BFD session globally, or at the interface level for specific interfaces only.

The following sections provide information about how to configure BFD support for BGP, IS-IS, OSPF, PIM, VRRP Tracking, and Static Routes:

"Configuring BFD Support for BGP" on page 22-18.

"Configuring BFD Support for IS-IS" on page 22-21

"Configuring BFD Support for OSPF" on page 22-23.

"Configuring BFD Support for PIM" on page 22-26

"Configuring BFD Support for VRRP Address Tracking" on page 22-30.

"Configuring BFD Support for Static Routes" on page 22-31.

Configuring BFD Support for BGP

The steps below show how to configure and verify BFD support for the BGP protocol, so that BGP is a registered protocol with BFD and receives forwarding path detection failure messages from BFD.

Note. BFD must be configured and enabled on the participating BGP interfaces. See "Configuring BFD Session Parameters" on page 22-14 for more information.

1 To associate BGP protocol with BFD liveliness detection, register BGP with BFD at the protocol level using the **bgp neighbor check-first-as** command as shown below:

```
-> ip bgp bfd-state enable
```

The BFD status for the BGP protocol is now enabled, which means that communication between BGP and BFD is enabled. To de-register BGP with BFD, enter the following command:

```
-> ip bgp bfd-state disable
```

To verify the BFD status for BGP protocol, you can use the **show ip bgp** command as shown below:

```
-> show ip bgp
Admin Status                          = disabled,
Operational Status                    = down,
Autonomous System Number              = 1,
BGP Router Id                         = 0.0.0.0,
Confederation Identifier              = 0,
IGP Synchronization Status            = disabled,
Minimum AS Origin Interval (seconds)  = 15,
Default Local Preference              = 100,
Route Reflection                      = disabled,
Cluster Id                            = 0.0.0.0,
Missing MED Status                    = Best,
Aspath Comparison                     = enabled,
Always Compare MED                    = disabled,
Fast External FailOver                = disabled,
Log Neighbor Changes                  = disabled,
Multiple Paths                        = disabled,
Graceful Restart                      = enabled,
Graceful Restart Status               = Not Restarting,
Configured Graceful Restart Interval = 90s,
IPv4 Unicast                          = enabled,
IPv6 Unicast                          = disabled,
BFD Status                            = disabled
ASN Output Format                     = asplain
```

2 Once BGP is registered with BFD at the protocol level, you need to enable BFD for particular IPv4 or IPv6 BGP neighbors using the **ip|ipv6 bgp neighbor bfd-state** command as shown below:

```
-> ip bgp neighbor 135.10.10.2 bfd-state enable
-> ipv6 bgp neighbor fe80::2efa:a2ff:fe13:e402 bfd-state enable
```

The above commands enable BFD for a neighbor with IPv4 address 135.10.10.2 and a neighbor with IPv6 address fe80::2efa:a2ff:fe13:e402. To enable BFD for all BGP neighbors (IPv4 and IPv6), use the **ip bgp bfd-state all-neighbors** command as shown below:

```
-> ip bgp bfd-state all-neighbors enable
```

To disable BFD for all configured BGP neighbors, use the **ip bgp bfd-state all-neighbors** with the **disable** keyword, as shown below:

```
-> ip bgp bfd-state all-neighbors disable
```

To display the BFD status of IPv4 BGP neighbors, use the **show ip bgp neighbors** command. For example:

```
-> show ip bgp neighbors
Legends: Nbr = Neighbor
         As  = Autonomous System
Nbr address     As    Admin state Oper state   BGP Id       Up/Down     BFD Status
---------------+-----+-----------+------------+-----------+------------+---------
100.1.1.10      2     enabled     established  3.3.3.3      00h:02m:19s  enabled
192.40.4.29     3     enabled     established  192.40.4.29  00h:14m:48s  disabled
192.40.4.121    5     disabled    idle         0.0.0.0      00h:00m:00s  enabled
```

To display the BFD status of IPv6 BGP neighbors, use the **show ipv6 bgp neighbors** command. For example:

```
-> show ipv6 bgp neighbors
Legends: Nbr = Neighbor
         As  = Autonomous System
Nbr address              As   Admin state Oper state  BGP Id   Up/Down      BFD
                                                                            Status
------------------------+----+-----------+-----------+--------+-----------+--------
2001:100:3:4::1          30   enabled     established 11.4.0.1 01h:42m:08s enabled
fe80::200:57ff:fe28:7e89 10   enabled     established 11.5.0.1 01h:40m:58s disabled
```

Use the **show ip|ipv6 bfd sessions** command to view BFD sessions with all BFD neighbors. For example:

```
-> show ip bfd sessions
Legends: Neg.     = Negotiated
         Discr    = Discriminator
         Intvl    = Interval (in milliseconds)
Local Interface    Neighbor                State   Remote  Neg. Rx Neg. Tx EchoRx
 Discr   Name       Address                         Discr   Intvl   Intvl   Intvl
------+---------+-----------------------+-------+-------+-------+-------+--------
 1      v1001     100.1.1.10               UP      0       0       0       ECHO
 2      v2000     101.1.1.11               UP      10      300     300     ASYNC

-> show ip bfd sessions 1
Local discriminator              = 1,
Neighbor IP Address              = 100.1.1.10,
Requested Session Type           = ECHO,
Interface IP Address             = 100.1.1.1,
Source UDP Port                  = 49152,
State                            = UP,
Session Operating Mode           = ECHO only,
Remote discriminator             = 0,
Negotiated Tx interval           = 0,
Negotiated Rx interval           = 0,
Echo Rx interval                 = 300,
Multiplier                       = 3,
Applications Registered:         = STATIC-ROUTING BGP

-> show ipv6 bfd sessions
Legends: Neg.     = Negotiated
         Discr    = Discriminator
         Intvl    = Interval (in milliseconds)
 Local Interface    Neighbor                State   Remote  Neg. Rx Neg. Tx EchoRx
 Discr   Name       Address                         Discr   Intvl   Intvl   Intvl
------+---------+-----------------------+-------+-------+-------+-------+--------
1      bfd-intf3 fe80::2efa:a2ff:fe13:e402  UP      0       0       0       300

-> show ipv6 bfd sessions 1
Local discriminator              = 1,
Neighbor IP Address              = fe80::2efa:a2ff:fe13:e402,
Requested Session Type           = ECHO,
Interface IP Address             = fe80::2efa:a2ff:fe13:e403,
Source UDP Port                  = 49152,
State                            = UP,
Session Operating Mode           = ECHO only,
Remote discriminator             = 0,
Negotiated Tx interval           = 0,
Negotiated Rx interval           = 0,
Echo Rx interval                 = 300,
Multiplier                       = 3,
Applications Registered:         = STATIC-ROUTING
```

Configuring BFD Support for IS-IS

BFD support for IS-IS is configured on a VLAN basis and is applied to all IPv4 and IPv6 interfaces associated with the VLAN. A single IS-IS adjacency covers both IPv4 and IPv6 interfaces, but the interfaces are treated independently within the adjacency. If an IS-IS adjacency has both interface types, there will be two BFD sessions (one for each interface). When one interface goes down, only the routes learned through that interface are removed.

The steps below show how to configure and verify BFD support for IS-IS, so that IS-IS is a registered protocol with BFD and receives forwarding path detection failure messages from BFD.

Note. IS-IS must be running on all participating routers, and BFD must be configured and enabled on the participating IS-IS VLANs. See "Configuring BFD Session Parameters" on page 22-14 for more information.

---

1 To associate BFD with the IS-IS protocol and to change the default BFD status for the IS-IS protocol, register IS-IS with BFD at the protocol level using the **ip isis bfd-state** command. For example:

```
-> ip isis bfd-state enable
```

The BFD status for the IS-IS protocol is now enabled, which means that communication between IS-IS and BFD is enabled. To de-register IS-IS with BFD, enter the following command:

```
-> ip isis bfd-state disable
```

2 To verify the BFD status for IS-IS protocol, use the **show ip isis status** command. For example:

```
-> show ip isis status
===============================================================================
ISIS Status
===============================================================================
System Id             : 2cfa.a213.e402
Admin State           : DOWN
Protocols Enabled     : IPv4 IPv6
Last Enabled          : Mon Oct 30 06:58:41 2017
Level Capability      : L1L2
Authentication Check  : True
Authentication Type   : None
Graceful Restart      : Disabled
GR helper-mode        : Disabled
LSP Lifetime          : 1200
LSP Wait              : Max: 5 sec  Initial: 0 sec  Second: 1 sec
Adjacency Check       : Loose
L1 Auth Type          : None
L2 Auth Type          : None
L1 Wide Metrics-only  : Disabled
L2 Wide Metrics-only  : Disabled
L1 LSDB Overload      : Disabled
L2 LSDB Overload      : Disabled
L1 LSPs               : 0
L2 LSPs               : 0
Last SPF              : Mon Oct 30 06:58:41 2017
SPF Wait              : Max: 10000 ms  Initial: 1000 ms  Second: 1000 ms
Hello-Auth Check      : Enabled
Csnp-Auth Check       : Enabled
Psnp-Auth Check       : Enabled
L1 Hello-Auth Check   : Enabled
L1 Csnp-Auth Check    : Enabled
```

```
L1 Psnp-Auth Check     : Enabled
L2 Hello-Auth Check    : Enabled
L2 Csnp-Auth Check     : Enabled
L2 Psnp-Auth Check     : Enabled
Multi-Topology         : Disabled
Auto-Configuration     : Disabled
Area Address           : None
BFD Status             : Disabled
```

3 Once IS-IS is registered with BFD at the protocol level, enable BFD on the participating IS-IS VLANs using the **ip isis vlan bfd-state** command. For example:

```
-> ip isis vlan 10 bfd-state enable
```

The above command enables BFD on IS-IS VLAN 10. To enable BFD on all IS-IS VLANs, use the **ip isis bfd-state all-vlans** command. For example:

```
-> ip isis bfd-state all-vlans enable
```

To disable BFD for all IS-IS VLANs, use the **ip isis bfd-state all-vlans** command with the **disable** keyword. For example:

```
-> ip isis bfd-state all-vlans disable
```

4 To display the BFD status on an IS-IS VLAN, use the **show ip isis vlan** command with the **detail** keyword. For example:

```
-> show ip isis vlan detail
============================================================================
ISIS Interface
============================================================================
----------------------------------------------------------------------------
VlanId              : 10             Level Capability  : L1L2
Oper State          : Up             Admin State       : Up
Auth Type           : Keychain(3)    Address Families  : IPv4, IPv6
Circuit Id          : 1              RetransmitInt     : 5
Type                : Broadcast      LSP Pacing Int    : 100
Mesh Group          : Inactive       CSNP Int          : 10
BFD Status    : Disabled

Level               : 1              Adjacencies       : 0
Desg IS             : abr_nyc
Auth Type           : None           Metric            : 10
Hello Timer         : 9              Hello Mult        : 3
Priority            : 64             Passive           : No
Level               : 2              Adjacencies : 0
Desg IS             : abr_nyc
Auth Type           : None            Metric           : 10
Hello Timer         : 9               Hello Mult       : 3
Priority            : 64              Passive          : No
```

Configuring BFD Support for OSPF

The steps below show how to configure and verify BFD support for OSPF and OSPFv3, so that OSPF and OSPFv3 are registered protocols with BFD and receive forwarding path detection failure messages from BFD.

Note. OSPF or OSPFv3 must be running on all participating routers, and BFD must be configured and enabled on the participating OSPF or OSPFv3 interfaces. See "Configuring BFD Session Parameters" on page 22-14 for more information.

1 To associate BFD with the OSPF or OSPFv3 protocol and to change the default BFD status for the OSPF or OSPFv3 protocol, register OSPF or OSPFv3 with BFD at the protocol level using the **ip ospf bfd-state** or the **ipv6 ospf bfd-state** command. For example:

```
-> ip ospf bfd-state enable
-> ipv6 ospf bfd-state enable
```

The BFD status for the OSPF and OSPFv3 protocol is now enabled, which means that communication between OSPF and BFD is enabled and between OSPFv3 and BFD is enabled. To de-register OSPF or OSPFv3 with BFD, enter the following commands:

```
-> ip ospf bfd-state disable
-> ipv6 ospf bfd-state disable
```

2 To verify the BFD status for the OSPF or OSPFv3 protocol, use the **show ip ospf** or the **show ipv6 ospf** command. For example:

```
-> show ip ospf
Router Id                        = 10.172.18.16,
OSPF Version Number              = 2,
Admin Status                     = Enabled,
Area Border Router ?             = No,
AS Border Router Status          = Disabled,
Route Tag                        = 0,
SPF Hold  Time (in seconds)      = 10,
SPF Delay  Time (in seconds)     = 5,
MTU Checking                     = Disabled,
# of Routes                      = 9,
# of AS-External LSAs            = 0,
# of self-originated LSAs        = 1,
# of LSAs received               = 0,
External LSDB Limit              = -1,
Exit Overflow Interval           = 0,
# of SPF calculations done       = 4,
# of Incr SPF calculations done  = 0,
# of Init State Nbrs             = 0,
# of 2-Way State Nbrs            = 0,
# of Exchange State Nbrs         = 0,
# of Full State Nbrs             = 0,
# of attached areas              = 1,
# of Active areas                = 1,
# of Transit areas               = 0,
# of attached NSSAs              = 0,
Default Route Origination        = none,
Default Route Metric-Type/Metric = type2 / 1
BFD Status                       = Disabled
Opaque Transit Capability        = Enabled
```

```
-> show ipv6 ospf
Status                          = Enabled,
Router ID                       = 30.1.1.2,
# Areas                         = 1,
# Interfaces                    = 3,
Area Border Router              = No,
AS Border Router                = No,
External Route Tag              = 0,
SPF Hold (seconds)              = 10,
SPF Delay (seconds)             = 5,
MTU checking                    = Enabled,
BFD Status                      = Disabled,
# SPF calculations performed    = 34,
Last SPF run (seconds ago)      = N/A,
# of routes                     = 1,
# of AS external LSAs           = 0,
# of neighbors that are in:
  Full state                    = 1,
  Loading state                 = 0,
  Exchange state                = 0,
  Exstart state                 = 0,
  2way state                    = 0,
  Init state                    = 0,
  Attempt state                 = 0,
  Down state                    = 0,
Restart Support                 = Enabled,
Restart Status                  = Restating,
Restart Helper Support          = Enabled,
Restart Helper Status           = NotHelpin
```

3 Once OSPF or OSPFv3 is registered with BFD at the protocol level, enable the OSPF or OSPFv3 interface(s) that participate in BFD using the **ip ospf interface bfd-state** or the **ipv6 ospf interface bfd-state** command. For example:

```
-> ip ospf interface vlan-10 bfd-state enable
-> ipv6 ospf interface vlan-2071 bfd-state enable
```

The above command enables BFD on the interfaces named vlan-10 and vlan-20. To enable BFD on all configured OSPF or OSPFv3 interfaces, use the **ip ospf bfd-state all-interfaces** or the **ipv6 ospf bfd-state all-interfaces** command. For example:

```
-> ip ospf bfd-state all-interfaces enable
-> ipv6 ospf bfd-state all-interfaces enable
```

To disable BFD for all configured OSPF or OSPFv3 interfaces, use the **ip ospf bfd-state all-interfaces** or the **ipv6 ospf bfd-state all-interfaces** command with the **disable** keyword. For example:

```
-> ip ospf bfd-state all-interfaces disable
-> ipv6 ospf bfd-state all-interfaces disable
```

4 To display the BFD status on an OSPF or OSPFv3 interface, use the **show ip ospf interface** or the **show ipv6 ospf interface** command. For example:

```
-> show ip ospf interface
   Interface          DR            Backup DR      Admin    Oper          BFD
     Name           Address          Address       Status   Status State  Status
-----------------+---------------+---------------+--------+------+------+-------
vlan-10          213.10.10.1     213.10.10.254   enabled   up     DR    enabled
vlan-20          215.10.10.254   215.10.10.1     enabled   up     BDR   disabled
```

```
-> show ipv6 ospf interface
                                                    IPv6
                                            Admin   Intf    Intf   Intf      BFD
 Name      DR Router ID  BDR Router ID     Status  Status  Type   State    Status
----------+-------------+--------------+---------+------+------+-------+--------
vlan-2071    5.5.5.5       0.0.0.0       Enabled   Up     BCAST   DR     Enabled
vlan-2055    7.7.7.7       5.5.5.5       Enabled   Up     BCAST   BDR    Enabled
vlan-2056    7.7.7.7       5.5.5.5       Enabled   Up     BCAST   BDR    Disabled
```

5 Once OSPF or OSPFv3 is registered with BFD at the protocol level and BFD is enabled on the desired OSPF or OSPFv3 interface(s), use the **show ip|ipv6 bfd interfaces** command to display BFD-enabled interfaces. For example:

```
-> show ip bfd interfaces
Interface  Admin    Tx        Min Rx   Min EchoRx Detect     OperStatus
Name       Status   Interval  Interval Interval   Multiplier
---------+--------+---------+---------+----------+----------+----------
bfd-intf1 enabled   300      300       300         3         UP
bfd-intf2 enabled   300      300       300         3         UP

-> show ipv6 bfd interfaces
    Interface          Admin         Tx      Min Rx    Min EchoRx Detect  Oper
      Name             Status     Interval  Interval   Interval    Mult  Status
---------------------+----------+----------+----------+----------+------+------
bfd-intf3             disabled    300        300        300        3      DOWN
```

6 To establish BFD sessions with neighbors that are in full state only, enter the **ip ospf interface bfd-state drs-only** or the **ipv6 ospf interface bfd-state drs-only** command as shown below:

```
-> ip ospf interface int1 bfd-state drs-only
-> ipv6 ospf interface int2 bfd-state drs-only
```

The above commands establish a BFD session on interface named int1 with OSPF DR neighbors in full state only and on interface named int2 with OSPFv3 DR neighbors in full state only.

To establish a BFD session on an interface with all neighbors which are greater than or equal to “2-way” state, use the **ip ospf interface bfd-state all-neighbors** or the **ipv6 ospf interface bfd-state all-neighbors** command as shown below:

```
-> ip ospf interface int2 bfd-state all-neighbors enable
-> ipv6 ospf interface int3 bfd-state all-neighbors enable
```

The above commands establish a BFD session on interface named int2 with all OSPF neighbors that are greater than or equal to “2-way” state and on interface named int3 with OSPFv3 neighbors that are greater than or equal to “2-way” state.

When any neighbors are added to either interface, OSPF informs BFD about the newly added neighbor(s); BFD then establishes a session with them. Use the **show ip|ipv6 bfd sessions** command to view BFD sessions with all BFD neighbors, as shown below:

```
-> show ip bfd sessions
Legends: Neg.     = Negotiated
         Discr    = Discriminator
         Intvl    = Interval (in milliseconds)
 Local   Interface  Neighbor       State     Remote   Neg. Rx   Neg. Tx  EchoRx
 Discr   Name       Address                  Discr    Intvl     Intvl    Intvl
------+-----------+-------------+----------+------+----------+--------+-------
 1      v1001       101.1.1.11       UP         1      300       300      300
 2      v2000       200.1.1.1        UP         0      0         0        300
```

```
-> show ipv6 bfd sessions
Legends: Neg.     = Negotiated
         Discr    = Discriminator
         Intvl    = Interval (in milliseconds)
 Local Interface    Neighbor                State   Remote  Neg. Rx Neg. Tx EchoRx
 Discr   Name       Address                         Discr   Intvl   Intvl   Intvl
------+---------+-----------------------+-------+-------+-------+-------+--------
1      bfd-intf3 fe80::2efa:a2ff:fe13:e402  UP      0       0       0       300
```

To view a BFD session with a particular neighbor, use the **show ip|ipv6 bfd sessions** command followed by the session number. For example:

```
-> show ip bfd sessions 1
Local discriminator              = 1,
Neighbor IP Address              = 101.1.1.11,
Requested Session Type           = ASYNC ,
Interface IfIndex                = 2,
Source UDP Port                  = 49153,
State                            = UP,
Session Operating Mode           = None,
Remote discriminator             = 1,
Negotiated Tx interval           = 300,
Negotiated Rx interval           = 300,
Echo Rx interval                 = 300,
Multiplier                       = 3,
Applications Registered:         = OSPF

-> show ipv6 bfd sessions 1
Local discriminator              = 1,
Neighbor IP Address              = fe80::2efa:a2ff:fe13:e402,
Requested Session Type           = ECHO,
Interface IP Address             = fe80::2efa:a2ff:fe13:e403,
Source UDP Port                  = 49152,
State                            = UP,
Session Operating Mode           = ECHO only,
Remote discriminator             = 0,
Negotiated Tx interval           = 0,
Negotiated Rx interval           = 0,
Echo Rx interval                 = 300,
Multiplier                       = 3,
Applications Registered:         = STATIC-ROUTING
```

Whenever there is any change to the interface/neighbor list or interface/neighbor state, OSPF immediately informs BFD about the changes. Additionally, whenever BFD detects any changes to the other end, BFD updates its database accordingly and informs OSPF for its fastest convergence.

Configuring BFD Support for PIM

The steps below show how to configure and verify BFD support for IPv4 and IPv6 PIM, so that both are registered protocols with BFD and receive forwarding path detection failure messages from BFD.

Note. PIM must be running on all participating routers, and BFD must be configured and enabled on the participating PIM interfaces. See "Configuring BFD Session Parameters" on page 22-14 for more information.

---

1 To associate BFD with the PIM protocol and to change the default BFD status for the PIM protocol, register IPv4 or IPv6 PIM with BFD at the protocol level using the **ip pim bfd-state** or the **ipv6 pim bfd-state** command. For example:

```
-> ip pim bfd-state enable
-> ipv6 pim bfd-state enable
```

The BFD status for IPv4 and IPv6 PIM is now enabled, which means that communication between PIM and BFD is enabled. To de-register IPv4 or IPv6 PIM with BFD, enter the following commands:

```
-> ip pim bfd-state disable
-> ipv6 pim bfd-state disable
```

2 Verify the BFD status for IPv4 or IPv6 PIM.

To verify the BFD status for IPv4 PIM, use the **show ip pim sparse** and **show ip pim dense** commands. For example:

```
-> show ipv6 pim sparse
Status                      = disabled,
Keepalive Period            = 210,
Max RPs                     = 32,
Probe Time                  = 5,
Register Suppress Timeout   = 60,
RP Switchover               = enabled,
SPT Status                  = enabled,
BIDIR Status                = disabled,
BIDIR Periodic Interval     = 60,
BIDIR DF Abort Status       = disabled,
BFD Status                  = disabled,
ASM Fast Join               = disabled,
SSM Fast Join               = disabled,
BIDIR Fast Join             = disabled,
BIDIR SSM Compatibility     = disabled
Register Rate Limit         = 100

-> show IPv6 pim dense
Status                        = enabled,
Source Lifetime               = 210,
State Refresh Interval        = 60,
State Refresh Limit Interval = 0,
State Refresh TTL             = 16
BFD Status                    = enabled
```

To verify the BFD status for IPv6 PIM, use the **show ipv6 pim sparse** and **show ipv6 pim dense** commands. For example:

```
-> show ipv6 pim sparse
Status                      = disabled,
Keepalive Period            = 210,
Max RPs                     = 32,
Probe Time                  = 5,
Register Suppress Timeout   = 60,
RP Switchover               = enabled,
SPT Status                  = enabled,
BIDIR Status                = disabled,
BIDIR Periodic Interval     = 60,
BIDIR DF Abort Status       = disabled,
BFD Status                  = disabled,
```

```
ASM Fast Join              = disabled,
SSM Fast Join              = disabled,
BIDIR Fast Join            = disabled,
BIDIR SSM Compatibility    = disabled
Register Rate Limit        = 100

-> show IPv6 pim dense
Status                       = enabled,
Source Lifetime              = 210,
State Refresh Interval       = 60,
State Refresh Limit Interval = 0,
State Refresh TTL            = 16
BFD Status                   = enabled
```

3 Once PIM is registered with BFD at the protocol level, enable the PIM interface(s) that participate in BFD using the **ip pim interface bfd-state** or the **ipv6 pim interface bfd-state** command. For example:

```
-> ip pim interface pimInt1 bfd-state enable
-> ipv6 pim interface pimInt2 bfd-state enable
```

The above command enables BFD on the IPv4 PIM interface named pimInt1 and the IPv6 PIM interface named pimInt2. To enable BFD on all configured PIM interfaces, use the **ip pim bfd-state all-interfaces** or the **ipv6 pim bfd-state all-interfaces** command. For example:

```
-> ip pim bfd-state all-interfaces enable
-> ipv6 pim bfd-state all-interfaces enable
```

To disable BFD for all configured PIM interfaces, use the **ip pim bfd-state all-interfaces** or the **ipv6 pim bfd-state all-interfaces** command with the **disable** keyword. For example:

```
-> ip pim bfd-state all-interfaces disable
-> ipv6 pim bfd-state all-interfaces disable
```

4 To display the BFD status on an IPv4 or IPv6 PIM interface, use the **show ip pim interface** or the **show ipv6 pim interface** command. For example:

```
-> show ip pim interface

Total 1 Interfaces

Interface Name    IP Address    Designated   Hello    J/P      Oper     BFD
                                Router       Interval Interval Status   Status
----------------+-------------+-------------+--------+--------+--------+--------
vlan-203         11.12.203.8   11.12.203.8   30       60       disabled disabled

-> show ipv6 pim interface

Total 3 Interfaces

Interface Name      Designated                Hello    J/P      Oper     BFD
                    Router                    Interval Interval Status   Status
------------------+-------------------------+--------+--------+--------+--------
vlan-5             fe80::2d0:95ff:feac:a537   30       60       enabled  disabled
vlan-30            fe80::2d0:95ff:feac:a537   30       60       disabled disabled
vlan-40            fe80::2d0:95ff:fee2:6eec   30       60       enabled  disabled
```

5 Once PIM is registered with BFD at the protocol level and BFD is enabled on the desired PIM interface(s), use the **show ip|ipv6 bfd interfaces** command to display BFD-enabled interfaces. For example:

```
-> show ip bfd interfaces
Interface  Admin    Tx        Min Rx   Min EchoRx Detect     OperStatus
Name       Status   Interval  Interval Interval   Multiplier
---------+--------+---------+---------+----------+----------+----------
bfd-intf1 enabled   300      300       300         3         UP
bfd-intf2 enabled   300      300       300         3         UP

-> show ipv6 bfd interfaces
    Interface          Admin         Tx      Min Rx    Min EchoRx Detect  Oper
      Name             Status     Interval  Interval   Interval    Mult  Status
---------------------+----------+----------+----------+----------+------+------
bfd-intf3             disabled    300        300        300        3      DOWN
```

When any neighbors are added to either interface, PIM informs BFD about the newly added neighbor(s); BFD then establishes a session with them. Use the **show ip|ipv6 bfd sessions** command to view BFD sessions with all BFD neighbors, as shown below:

```
-> show ip bfd sessions
Legends: Neg.     = Negotiated
         Discr    = Discriminator
         Intvl    = Interval (in milliseconds)
 Local   Interface  Neighbor       State     Remote   Neg. Rx   Neg. Tx  EchoRx
 Discr   Name       Address                  Discr    Intvl     Intvl    Intvl
------+-----------+--------------+----------+------+----------+--------+-------
 1      v1001       101.1.1.11       UP         1      300       300      300
 2      v2000       200.1.1.1        UP         0      0         0        300

-> show ipv6 bfd sessions
Legends: Neg.     = Negotiated
         Discr    = Discriminator
         Intvl    = Interval (in milliseconds)
 Local Interface    Neighbor                 State   Remote  Neg. Rx Neg. Tx EchoRx
 Discr   Name       Address                          Discr   Intvl   Intvl   Intvl
------+---------+-------------------------+-------+-------+-------+-------+--------
1      bfd-intf3 fe80::2efa:a2ff:fe13:e402  UP      0       0       0       300
```

To view a BFD session with a particular neighbor, use the **show ip|ipv6 bfd sessions** command followed by the session number. For example:

```
-> show ip bfd sessions 1

Local discriminator              = 1,
Neighbor IP Address              = 101.1.1.11,
Requested Session Type           = ASYNC ,
Interface IfIndex                = 2,
Source UDP Port                  = 49153,
State                            = UP,
Session Operating Mode           = None,
Remote discriminator             = 1,
Negotiated Tx interval           = 300,
Negotiated Rx interval           = 300,
Echo Rx interval                 = 300,
Multiplier                       = 3,
Applications Registered:         = OSPF

-> show ipv6 bfd sessions 1
Local discriminator              = 1,
Neighbor IP Address              = fe80::2efa:a2ff:fe13:e402,
Requested Session Type           = ECHO,
```

```
Interface IP Address              = fe80::2efa:a2ff:fe13:e403,
Source UDP Port                   = 49152,
State                             = UP,
Session Operating Mode            = ECHO only,
Remote discriminator              = 0,
Negotiated Tx interval            = 0,
Negotiated Rx interval            = 0,
Echo Rx interval                  = 300,
Multiplier                        = 3,
Applications Registered:          = STATIC-ROUTING
```

Whenever there is any change to the interface/neighbor list or interface/neighbor state, PIM immediately informs BFD about the changes. Additionally, whenever BFD detects any changes to the other end, BFD updates its database accordingly and informs PIM for its fastest convergence.

Configuring BFD Support for VRRP Address Tracking

The steps below show you how to configure and verify BFD support for VRRP protocol, so that VRRP is a registered protocol with BFD and receives forwarding path detection failure messages from BFD. Once VRRP is a registered protocol with BFD, then BFD can be enabled for a specific VRRP address tracking policy.

1 To associate VRRP protocol with BFD liveliness detection, register VRRP with BFD at the protocol level using the **vrrp bfd-state** command as shown below:

```
-> vrrp bfd-state enable
```

Note. VRRP protocol supports BFD in the echo-only operational mode.

BFD status for VRRP protocol is now enabled, which means that socket communication between VRRP and BFD is enabled.

To de-register VRRP with BFD, enter the following command at the system prompt:

```
-> vrrp bfd-state disable
```

To verify the BFD status for VRRP protocol, you can use the **show vrrp** command as shown below:

```
-> show vrrp
VRRP default advertisement interval: 5 seconds
VRRP default priority: 100
VRRP default preempt: Yes
VRRP trap generation: Enabled
VRRP startup delay: 45  (expired)
VRRP BFD-STATUS : Enabled

               IP         Admin                       Adv.
VRID VLAN   Address(es)   Status   Priority Preempt Interval
----+----+---------------+--------+--------+------+----------
   1  101 192.60.245.240  Enabled      100    Yes        5
   2  102 192.60.246.240  Enabled      100    Yes        5

-> show vrrp3
VRRP trap generation: Enabled
VRRP startup delay: 50  (expired)
VRRP BFD-STATUS : Enabled
```

```
                                        Admin                                Adv.
VRID VLAN     IPv6 Address(es)          Status  Priority Preempt Accept Interval
----+----+---------------------------+--------+--------+-------+------+--------
  1  101    fe80::200:5eff:fe00:201  Enabled    200      No    Yes      100
            1010::30
  2  102    fe80::200:5eff:fe00:202  Enabled    200      No    Yes      100
            1020::30
```

2 Once VRRP is registered with BFD at the protocol level, enable BFD for a particular VRRP address tracking policy using the **vrrp track** command. Ensure that the track policy is associated with at least one of the virtual routers. For example:

```
-> vrrp track 2 address 192.60.245.240 bfd-state enable
-> vrrp track 5 address fe80::200:5eff:fe00:202 bfd-state enable
```

The above commands enable BFD for an IPv4 and IPv6 address tracking policy (VRRP track number 2 and 5) to track remote interface address 192.60.245.240 and fe80::200:5eff:fe00:202.

Notes:

- The value of the address parameter should be a remote interface address. BFD cannot be configured for a local interface address.
- Enabling BFD for an address tracking policy requires a Loopback0 interface on the local switch. The IP address of this interface will serve as the source IP address of BFD packets. For more information about configuring a Loopback0 interface, see the “IP Commands” or “IPv6 Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Use the **show vrrp track** command to verify whether BFD is enabled for a particular track policy. For example:

```
-> show vrrp track
Track                                              Admin    Oper          BFD
 ID                   Policy                       State    State  Pri   Status
-----+---------------------------------------+----------+------+-----+---------
  2    192.60.245.240                          Enabled    Up      25   Enabled
  5    fe80::200:5eff:fe00:202                 Enabled    Up      25   Enabled
```

Use the **show ip|ipv6 bfd interfaces** command to verify the BFD interface/session configuration and operation status.

Once the track policy is configured, the BFD session is established with the remote IPv4 or IPv6 address. BFD session is also established with the BFD neighbors.

Use the **show ip|ipv6 bfd sessions** command to view BFD sessions with all BFD neighbors.

Configuring BFD Support for Static Routes

This section provides information about how to configure and verify BFD support for IPv4 and IPv6 static routing.

To change the default BFD status for a particular static route and to enable BFD support, use the **ip static-route bfd-state** or the **ipv6 static-route bfd-state** command. For example:

```
-> ip static-route 10.1.1.1 mask 255.0.0.0 gateway 10.1.1.25 bfd-state enable
```

```
-> ipv6 static-route 195:35::/64 gateway fe80::2d0:95ff:fe12:f470 bfd-state
enable
```

Note. Static Routes support BFD in the echo-only operational mode.

The above commands enable BFD support for an IPv4 static route (destination IP address as 10.1.1.1, destination network mask as 255.0.0.0, and gateway address as 10.1.1.25) and an IPv6 static route (destination IPv6 address 195:35::/64 and gateway address fe80::2d0:95ff:fe12:f470.

In order to create a BFD session for a static route, the gateway address should not match with any local interface address of the switch. If multiple routes are configured with the same gateway address, only one BFD session is run. To verify the BFD session list, which shows the gateway address, use the **show ip|ipv6 bfd sessions** command.

To enable BFD support for all static routes, use the **ip static-route all bfd-state** or the **ipv6 static-route all bfd-state** command:

```
-> ip static-route all bfd-state enable
-> ipv6 static-route all bfd-state enable
```

To verify the static routes on which BFD is enabled, use the **show ip router database** or the **show ipv6 router database** command with the **protocol static** option. For example:

```
-> show ip router database protocol static
Legend: + indicates routes in-use
        b indicates BFD-enabled static route
        i indicates interface static route
        r indicates recursive static route, with following address in brackets

Destination         Gateway         Interface    Protocol  Metric  Tag   Misc-Info
-------------------+---------------+------------+--------+-------+-----+-----------
+b 10.1.1.1/8         10.1.1.25       v1001        STATIC    1     0

Inactive Static Routes
Destination       Gateway        Metric    Tag   Misc-Info
----------------+--------------+--------+-------+-----------
1.0.0.0/8         8.4.5.3         1         0

-> show ipv6 router database protocol static
Legend: + indicates routes in-use
        b indicates BFD-enabled static route

Destination/Prefix Gateway Address            Interface       Metric    Tag   Misc-Info
------------------+-------------------------+---------------+--------+-----+---------
+b 195:35::/64     fe80::2d0:95ff:fe12:f470  v6if-6to4-137    1        0

Inactive Static Routes:
Vlan  Destination/Prefix Gateway Address          Metric    Tag   Misc-Info
-----+------------------+-----------------------+--------+-----+----------
  -    3333::/24          4444::                    1          0
```

BFD Application Example

This section provides an example network configuration in which BFD is associated with the OSPF protocol running on the network. In addition, a tutorial is also included that provides steps on how to configure the example network topology using the Command Line Interface (CLI).

Example Network Overview

The diagram below represents a simple OSPF network consisting of three routers. On all three routers, OSPF is associated with BFD for faster failure detection of any router on the network.

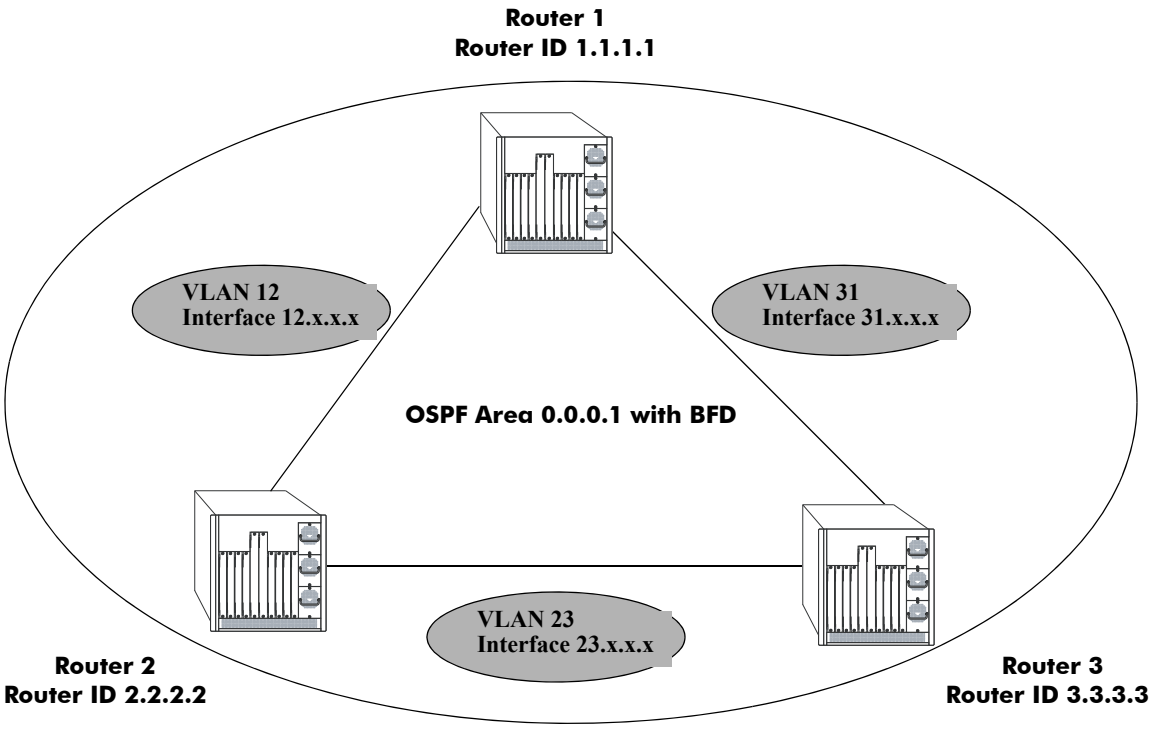


Figure 22-1 : Example OSPF Network using the BFD Protocol

The following steps are used to configure the example BFD-enabled OSPF network as shown in the diagram above.

Note. Configuring a BFD session explicitly with an IP interface name on individual routers is optional, and must be used if user defined BFD session parameters need to be applied. All the steps for explicit configuration are mentioned as optional.

Step 1: Prepare the Routers

The first step is to create the VLANs on each router, add an IP interface to the VLAN, assign a port to the VLAN, and assign a router identification number to the routers. For the backbone connection, the network design in this case uses slot 2, port 1 as the egress port and slot 2, port 2 as ingress port on each router. Router 1 connects to Router 2, Router 2 connects to Router 3, and Router 3 connects to Router 1.

Note. The ports are statically assigned to the router VLANs, as a VLAN must have a physical port assigned to it in order for the IP router interface to function.

The commands to set up the VLAN configuration are shown below:

Router 1 (using ports 2/1 and 2/2 for the backbone and ports 2/3-5 for end devices):

```
-> vlan 31
-> ip interface vlan-31 vlan 31 address 31.0.0.1 mask 255.0.0.0
-> vlan 31 members port 2/1

-> vlan 12
-> ip interface vlan-12 vlan 12 address 12.0.0.1 mask 255.0.0.0
-> vlan 12 members port 2/2

-> vlan 10
-> ip interface vlan-10 vlan 10 address 10.0.0.1 mask 255.0.0.0
-> vlan 10 members port 2/3-5

-> ip router router-id 1.1.1.1
```

These commands created VLANs 31, 12, and 10.

- VLAN 31 handles the backbone connection from Router 1 to Router 3, using the IP router port 31.0.0.1 and physical port 2/1.
- VLAN 12 handles the backbone connection from Router 1 to Router 2, using the IP router port 12.0.0.1 and physical port 2/2.
- VLAN 10 handles the device connections to Router 1, using the IP router port 10.0.0.1 and physical ports 2/3-5. More ports could be added at a later time if necessary.

The router was assigned the Router ID of 1.1.1.1.

Router 2 (using ports 2/1 and 2/2 for the backbone and ports 2/3-5 for end devices):

```
-> vlan 12
-> ip interface vlan-12 vlan 12 address 12.0.0.2 mask 255.0.0.0
-> vlan 12 members port 2/1

-> vlan 23
-> ip interface vlan-23 vlan 23 address 23.0.0.2 mask 255.0.0.0
-> vlan 23 members port 2/2

-> vlan 20
-> ip interface vlan-20 vlan 20 address 20.0.0.2 mask 255.0.0.0
-> vlan 20 members port 2/3-5

-> ip router router-id 2.2.2.2
```

These commands created VLANs 12, 23, and 20.

- VLAN 12 handles the backbone connection from Router 1 to Router 2, using the IP router port 12.0.0.2 and physical port 2/1.
- VLAN 23 handles the backbone connection from Router 2 to Router 3, using the IP router port 23.0.0.2 and physical port 2/2.
- VLAN 20 handles the device connections to Router 2, using the IP router port 20.0.0.2 and physical ports 2/3-5. More ports could be added at a later time if necessary.

The router was assigned the Router ID of 2.2.2.2.

Router 3 (using ports 2/1 and 2/2 for the backbone, and ports 2/3-5 for end devices):

```
-> vlan 23
-> ip interface vlan-23 vlan 23 address 23.0.0.3 mask 255.0.0.0
-> vlan 23 members port 2/1

-> vlan 31
-> ip interface vlan-31 vlan 31 address 31.0.0.3 mask 255.0.0.0
-> vlan 31 members port 2/2

-> vlan 30
-> ip interface vlan-30 vlan 30 address 30.0.0.3 mask 255.0.0.0
-> vlan 30 members port 2/3-5

-> ip router router-id 3.3.3.3
```

These commands created VLANs 23, 31, and 30.

- VLAN 23 handles the backbone connection from Router 2 to Router 3, using the IP router port 23.0.0.3 and physical port 2/1.
- VLAN 31 handles the backbone connection from Router 3 to Router 1, using the IP router port 31.0.0.3 and physical port 2/2.
- VLAN 30 handles the device connections to Router 3, using the IP router port 30.0.0.3 and physical ports 2/3-5. More ports could be added at a later time if necessary.

The router was assigned the Router ID of 3.3.3.3.

Step 2: Enable OSPF

The next step is to load and enable OSPF on each router. The commands for this step are below (the commands are the same on each router):

```
-> ip load ospf
-> ip ospf admin-state enable
```

Step 3: Create the OSPF Area

Now the area should be created. In this case, we create area 0.0.0.1. The command for this step is below (the command is the same on each router):

```
-> ip ospf area 0.0.0.1
```

Area 0.0.0.1 is created and enabled.

Step 4: Configure OSPF Interfaces

Next, OSPF interfaces must be created, enabled, and assigned to area 0.0.0.1. The OSPF interfaces should have the same interface name as the IP router interfaces created above in "Step 1: Prepare the Routers" on page 22-34.

Router 1

```
-> ip ospf interface vlan-31
-> ip ospf interface vlan-31 area 0.0.0.0
-> ip ospf interface vlan-31 admin-state enable

-> ip ospf interface vlan-12
-> ip ospf interface vlan-12 area 0.0.0.0
-> ip ospf interface vlan-12 admin-state enable

-> ip ospf interface vlan-10
-> ip ospf interface vlan-10 area 0.0.0.1
-> ip ospf interface vlan-10 admin-state enable
```

Router 2

```
-> ip ospf interface vlan-12
-> ip ospf interface vlan-12 area 0.0.0.0
-> ip ospf interface vlan-12 admin-state enable

-> ip ospf interface vlan-23
-> ip ospf interface vlan-23 area 0.0.0.0
-> ip ospf interface vlan-23 admin-state enable

-> ip ospf interface vlan-20
-> ip ospf interface vlan-20 area 0.0.0.2
-> ip ospf interface vlan-20 admin-state enable
```

Router 3

```
-> ip ospf interface vlan-23
-> ip ospf interface vlan-23 area 0.0.0.0
-> ip ospf interface vlan-23 admin-state enable

-> ip ospf interface vlan-31
-> ip ospf interface vlan-31 area 0.0.0.0
-> ip ospf interface vlan-31 admin-state enable

-> ip ospf interface vlan-30
-> ip ospf interface vlan-30 area 0.0.0.3
-> ip ospf interface vlan-30 admin-state enable
```

Step 5: (Optional) Configure BFD Interfaces

Next, BFD interfaces must be created and enabled. The BFD interfaces should have the same interface name as the IP router interfaces created above in "Step 1: Prepare the Routers" on page 22-34.

Router 1

```
-> ip bfd interface vlan-31
-> ip bfd interface vlan-31 admin-state enable

-> ip bfd interface vlan-12
-> ip bfd interface vlan-12 admin-state enable
```

```
-> ip bfd interface vlan-10
-> ip bfd interface vlan-10 admin-state enable
```

Router 2

```
-> ip bfd interface vlan-12
-> ip bfd interface vlan-12 admin-state enable

-> ip bfd interface vlan-23
-> ip bfd interface vlan-23 admin-state enable

-> ip bfd interface vlan-20
-> ip bfd interface vlan-20 admin-state enable
```

Router 3

```
-> ip bfd interface vlan-23
-> ip bfd interface vlan-23 admin-state enable

-> ip bfd interface vlan-31
-> ip bfd interface vlan-31 admin-state enable

-> ip bfd interface vlan-30
-> ip bfd interface vlan-30 admin-state enable
```

Step 6: (Optional) Configure Global BFD Parameters

Global BFD parameter settings for timer values and operational mode are applied to all BFD interfaces configured on the routing instance. When a BFD interface is created, the global settings are also applied as the default parameter values for the interface.

The following steps change the default global BFD parameter values for the example network; the commands used are the same on each router.

- Set the minimum amount of time BFD waits between each transmission of control packets to 200.

  `-> ip bfd transmit 200` milliseconds

- Set the minimum amount of time BFD waits to receive control packets to 200 milliseconds.

  ```
  -> ip bfd receive 200
  ```

- Set the global BFD Echo packet time interval to 200 milliseconds.

  ```
  -> ip bfd echo-interval 200
  ```

Step 7: Enable BFD and register OSPF with BFD

Once all the global BFD parameters are configured, enable BFD on all interfaces, register BFD with OSPF, and then enable BFD on all OSPF interfaces. The following steps are the same on each router:

'In this example, global BFD parameters will be used for the BFD sessions. Enable BFD admin status and register OSPF with BFD and then enable BFD on all OSPF interfaces. Repeat the following steps on each router:

```
-> ip bfd admin-state enable
-> ip ospf bfd-state enable
-> ip ospf bfd-state all-interfaces enable
```

Step 8: Examine the Network

After the network has been created, use the following **show** commands to check various aspects of the example network:

- To verify the configured BFD status on routers, use the **show ip bfd** command. This command shows the protocols registered for BFD (OSPF in example network) and the parameter values for the transmit, receive, and echo intervals, the multiplier number, and the operational mode.
- To display information about BFD sessions, use the **show ip|ipv6 bfd sessions** command.
- To check the BFD status at the OSPF protocol level, use the **show ip ospf** command. This command is also used to check the general OSPF configuration. For OSPF interfaces, use the **show ip ospf interface** command.

Verifying the BFD Configuration

To display information such as the BFD status for different session parameters and Layer 3 protocols, use the **show** commands listed in the following table:

show ip bfd	Displays the global BFD configuration for the routing instance.
show ip|ipv6 bfd interfaces	Displays the BFD interface configuration for the switch.
show ip|ipv6 bfd sessions	Displays the BFD neighbors and session states.
show ip bgp	Displays the BFD status for the BGP protocol.
show ip bgp neighbors	Displays the BFD status for IPv4 BGP neighbors.
show ipv6 bgp neighbors	Displays the BFD status for IPv6 BGP neighbors.
show ip isis status	Displays the BFD status for the IS-IS protocol.
show ip isis vlan	Displays the BFD status for IS-IS VLANs.
show ip ospf	Displays the BFD status for the OSPF protocol.
show ip ospf interface	Displays the BFD status for OSPF interfaces.
show ipv6 ospf	Displays the BFD status for the OSPFv3 protocol.
show ipv6 ospf interface	Displays the BFD status for OSPFv3 interfaces.
show ip pim sparse **show ip pim dense**	Displays the BFD status for the IPv4 PIM.
show ipv6 pim sparse **show ipv6 pim dense**	Displays the BFD status for the IPv6 PIM.
show ip pim interface	Displays the BFD status for the IPv4 PIM interface.
show ipv6 pim interface	Displays the BFD status for the IPv6 PIM interface.
show vrrp	Displays the BFD status for the VRRP and VRRP3 protocol.
show vrrp track	Displays the BFD status for a track policy.
show ip router database protocol static	Displays the BFD status for IPv4 static routes.
show ipv6 router database	Displays the BFD status for IPv6 static routes.

For more information about the resulting displays form these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. Examples of the above commands and their outputs are given in the section "Configuring BFD" on page 22-14.

23 Configuring DHCP Relay

The User Datagram Protocol (UDP) is a connectionless transport protocol that runs on top of IP networks. The DHCP Relay allows you to use nonroutable protocols (such as UDP) in a routing environment. UDP is used for applications that do not require the establishment of a session and end-to-end error checking. Email and file transfer are two applications that could use UDP. UDP offers a direct way to send and receive datagrams over an IP network and is primarily used for broadcasting messages. This chapter describes the DHCP Relay feature. This feature allows UDP broadcast packets to be forwarded across VLANs that have IP routing enabled.

In This Chapter

This chapter describes the basic components of DHCP Relay and how to configure them. CLI commands are used in the configuration examples. For more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following information and procedures are included in this chapter:

- "Quick Steps for Setting Up DHCP Relay" on page 23-3.
- "DHCP Relay Overview" on page 23-4.
- "Configuring DHCP Relay" on page 23-8.
- "Setting the DHCP Relay Forwarding Mode" on page 23-8.
- "Configuring DHCP Relay Parameters" on page 23-9.
- "Configuring the Status of the DHCP Relay Feature" on page 23-8.
- "Configuring the DHCP Client Interface" on page 23-13.
- "Configuring Generic UDP Relay" on page 23-17.
- "Configuring DHCP Security Features" on page 23-21.
- "Using the Relay Agent Information Option (Option-82)" on page 23-21.
- "Using DHCP Snooping" on page 23-24.
- "DHCPv6 Relay Overview" on page 23-32.
- "Quick Steps for Configuring DHCPv6 Relay" on page 23-32.
- "Configuring DHCPv6 Relay" on page 23-33.
- "Using DHCPv6 Snooping" on page 23-35.

For information about the IP protocol, see Chapter 17, "Configuring IP."

DHCP Relay Defaults

When the IP DHCP Relay feature is enabled using the **ip dhcp relay admin-state** command, the following default DHCP Relay parameter settings apply:

Parameter Description	Command	Default Value/Comments
Default UDP service	**ip udp relay service**	BOOTP/DHCP
Forward delay time value for DHCP Relay	**ip dhcp relay forward-delay**	0 seconds
Maximum number of hops	**ip dhcp relay maximum-hops**	16 hops
Packet forwarding option	**ip dhcp relay per-interface-mode**	Global mode
Relay Agent Information Option	**ip dhcp relay insert-agent-information**	Disabled
Automatic switch IP configuration for default VLAN 1	**ip interface dhcp-client**	Disabled

Quick Steps for Setting Up DHCP Relay

Configure DHCP Relay on switches where packets are routed between IP networks. The DHCP Relay feature is disabled by default. To set up DHCP Relay, proceed as follows:

1 Enable DHCP Relay for the switch using the **ip dhcp relay admin-state** command. For example:

```
-> ip dhcp relay admin-state enable
```

2 By default, a global DHCP Relay agent is active when the DHCP Relay feature is enabled for the first time. To configure a next hop destination IP address to which all DHCP packets are relayed, use the **ip dhcp relay destination** command. For example:

```
-> ip dhcp relay destination 128.100.16.1
```

3 To change the DHCP Relay agent forwarding mode, use the **ip dhcp relay per-interface-mode** command. For example, the following command enables the per-interface relay agent mode for the switch:

```
-> ip dhcp relay per-interface-mode
```

4 When the per-interface DHCP Relay agent mode is active, use the **ip dhcp relay interface destination** command to configure a next-hop destination IP address for a specific IP interface. For example:

```
-> ip dhcp relay interface ipv4-v200 destination 128.100.16.1
```

5 The DHCP forward delay time and maximum hop count values are global settings that apply regardless of which DHCP Relay agent mode is active (global or per-interface). By default, the forward delay time is set to zero seconds and the maximum hops count is set to 16. To change these values, use the **ip dhcp relay forward-delay** and **ip dhcp relay maximum-hops** commands. For example,

```
-> ip dhcp relay forward-delay 30
-> ip dhcp relay maximum-hops 10
```

6 To include local relay agent information into client DHCP packets when the agent forwards these packets to the destination IP address, use the **ip dhcp relay insert-agent-information** command. For example:

```
-> ip dhcp relay insert-agent-information
```

Note. Optional. To verify the DHCP Relay configuration, enter the **show ip dhcp relay interface** command. The display shown for the DHCP Relay configured in the above Quick Steps is shown here:

```
-> show ip dhcp relay interface
IP DHCP Relay :
  DHCP Relay Admin Status        = Enabled,
  Forward Delay(seconds)         = 30,
  Max number of hops             = 10,
  Relay Agent Information        = Enabled,
  Relay Agent Information Policy = Drop,
  DHCP Relay Opt82 Format        = Base MAC,
  DHCP Relay Opt82 String        = 00:e0:b1:e7:09:a3,
  PXE support                    = Disabled,
  Relay Mode                     = Global,
  Bootup Option                  = Disable,
  Relay Destination list (Global Mode):
    From Interface Any to Server 128.100.16.1
```

For more information about this display, see the “DHCP Relay Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

DHCP Relay Overview

The DHCP Relay service, its corresponding port numbers, and configurable options are as follows:

- DHCP Relay Service: BOOTP/DHCP
- UDP Port Numbers 67/68 for Request/Response
- Configurable options: DHCP server IP address, Forward Delay, Maximum Hops, Forwarding Option, automatic switch IP configuration

The port numbers indicate the destination port numbers in the UDP header. The DHCP Relay verifies whether the forward delay time (specified by the user) has elapsed before sending the packet down to UDP with the destination IP address replaced by the address (also specified by the user).

If the relay is configured with multiple IP addresses, then the packet is sent to all IP address destinations. The DHCP Relay also verifies that the maximum hop count has not been exceeded. If the forward delay time is *not* met or the maximum hop count is exceeded, the BOOTP/DHCP packet is discarded by the DHCP Relay.

The forwarding option allows you to specify if the relay must operate in the global and per-interface mode. The global mode forwards all DHCP packets on a global relay service. The per-interface mode forwards DHCP packets on an IP interface relay service. See "Setting the DHCP Relay Forwarding Mode" on page 23-8 for more information.

An additional function provided by the DHCP Relay service enables automatic IP address configuration for default VLAN 1 when an unconfigured switch boots up. If this function is enabled, the switch broadcasts a BootP or a DHCP request packet at boot time. When the switch receives an IP address from a BootP/DHCP server, the address is assigned to default VLAN 1. See "Enabling the DHCP Client Interface" on page 23-13 for more information.

Alternately, the relay function can be provided by an external router connected to the switch; in this case, the relay is configured on the external router.

DHCP

DHCP (Dynamic Host Configuration Protocol) provides a framework for passing configuration information to Internet hosts on a TCP/IP network. It is based on the Bootstrap Protocol (BOOTP), adding the ability to automatically allocate reusable network addresses and additional configuration options. DHCP consists of the following two components:

- A protocol for delivering host-specific configuration parameters from a DHCP server to a host.
- A mechanism for allocating network addresses to hosts.

DHCP is built on a client-server model in which a designated DHCP server allocates network addresses and delivers configuration parameters to dynamically configured hosts. It supports the following three mechanisms for IP address allocation.

Automatic—DHCP assigns a permanent IP address to a host.

Dynamic—DHCP assigns an IP address to a host for a limited period of time (or until the host explicitly relinquishes the address).

Manual—The network administrator assigns a host IP address and DHCP simply conveys the assigned address to the host.

DHCP and the OmniSwitch

The unique characteristics of the DHCP protocol require a good plan before setting up the switch in a DHCP environment. Since DHCP clients initially have no IP address, placement of these clients in a VLAN is hard to determine.

The DHCP feature on OmniSwitch provides two services to the network users:

- DHCP Relay Agent
- Generic UDP Relay

The DHCP Relay Agent provides the network interfaces dynamic IP addresses from the DHCP server present on a different VLAN. This feature can be configured using the **ip dhcp relay** and related commands for the VLAN domain and for the service domain. In the VLAN domain, the agent relays packets between the client and the server across VLANs. In the service domain, the agent relays packets between the client and the server across Shortest Path Bridging (SPB) services.

Generic UDP Relay forwards packets with pre-configured destination UDP port information to destination VLANs, SPB services, or an IP address. This feature is configured using the **ip udp relay** and related commands.

For more information on the CLI commands related to DHCP Relay, see the “DHCP Relay Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

External DHCP Relay Application

The DHCP Relay can be configured on a router that is external to the switch. In this application example the switched network has a single VLAN configured with multiple segments. All of the network hosts are DHCP-ready, meaning they obtain their network address from the DHCP server. The DHCP server resides behind an external network router that supports the DHCP Relay functionality.

The router must support DHCP Relay functionality to be able to forward DHCP frames. In this example, DHCP Relay is supported within an external router that forwards request frames from the incoming router port to the outgoing router port attached to the OmniSwitch.

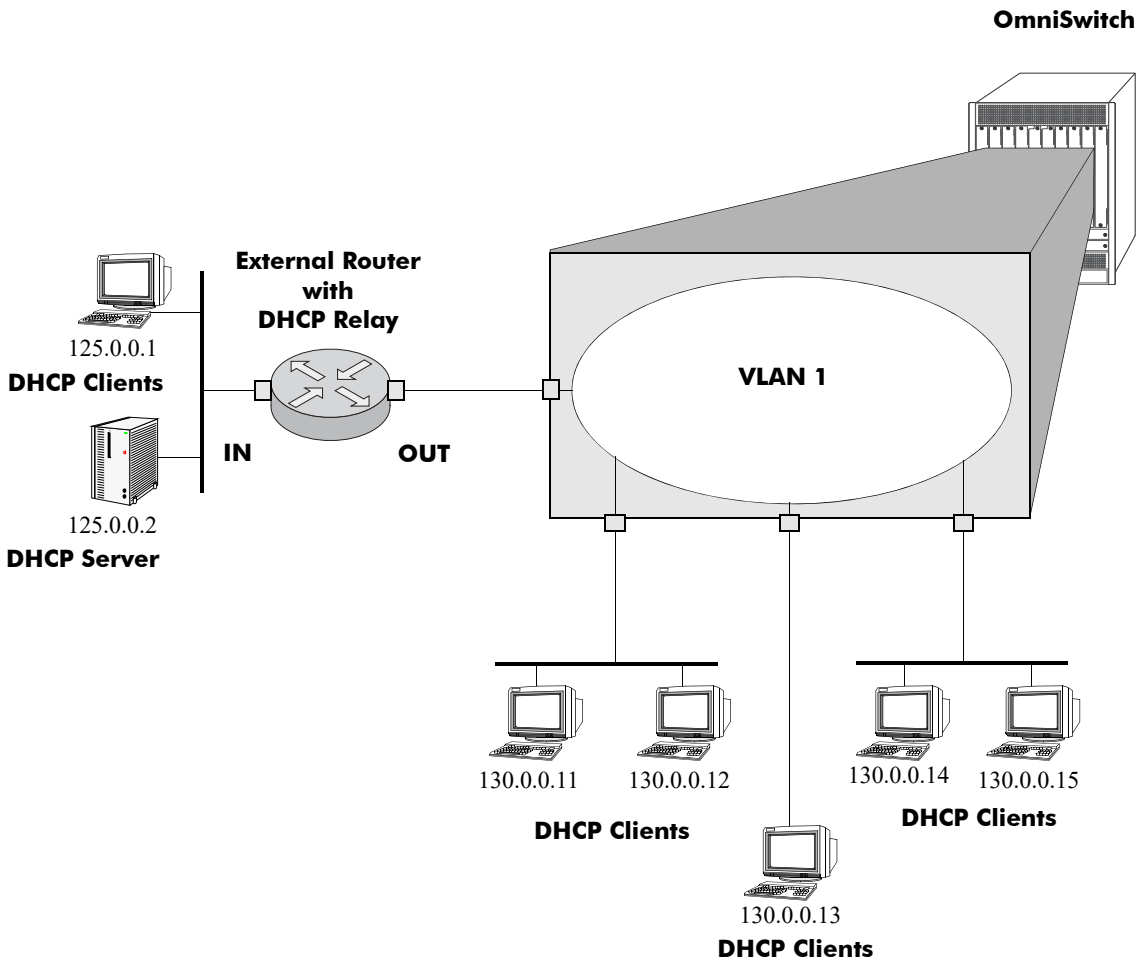


Figure 23-1 : DHCP Clients are Members of the Same VLAN

The external router inserts the subnet address of the first hop segment into the DHCP request frames from the DHCP clients. This subnet address allows the DHCP server to locate the segment on which the requesting client resides. In this example, all clients attached to the OmniSwitch are DHCP-ready and have the same subnet address (130.0.0.0) inserted into each of the requests by the DHCP Relay function of the router. The DHCP server assigns a different IP address to each of the clients. The switch does not need an IP address assigned to it. All DHCP clients are members of either a default VLAN or an IP protocol VLAN.

Internal DHCP Relay

The internal DHCP Relay is configured using the UDP forwarding feature in the switch, available through the **ip dhcp relay admin-state** command. For more information, see "Configuring DHCP Relay" on page 23-8.

This application example shows a network with two VLANs, each with multiple segments. All network clients are DHCP-ready and the DHCP server resides on just one of the VLANs. This example is much like the first application example, except that the DHCP Relay function is configured inside the switch. (See "Configuring DHCP Relay for the SPB Service Domain" on page 23-10 for an application example of configuring a relay agent for a service-bound IP interface.)

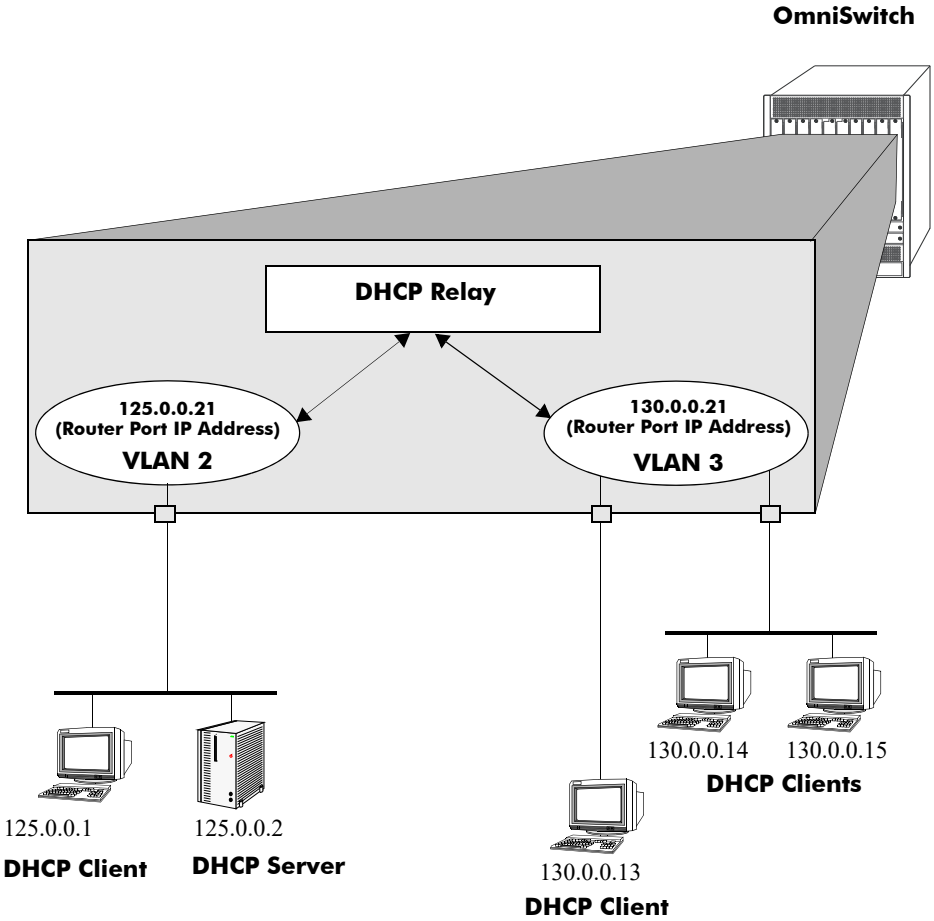


Figure 23-2 : DHCP Clients in Two VLANs

During initialization, each network client forwards a DHCP request frame to the DHCP server using the local broadcast address. For these locally attached stations, the frame is simply switched from one station to another.

In this case, the DHCP server and clients must be members of the same VLAN (they can also be members of the default VLAN).

Since the clients in the application example are not members of the same VLAN as the DHCP server, they must request an IP address through the DHCP Relay routing entity in the switch. When a DHCP request frame is received by the DHCP Relay entity, it is forwarded from VLAN 3 to VLAN 2. All the DHCP-ready clients in VLAN 3 must be members of the same VLAN, and the switch must have the DHCP Relay function configured.

Configuring DHCP Relay

A DHCP Relay agent can be configured to relay DHCP packets between a client and a DHCP server over a VLAN or Shortest Path Bridging (SPB) domain. The next hop IP address or the DHCP server IP address is specified as the destination to which client packets are forwarded. DHCP Relay is configured on switches where packets are routed between IP networks.

There are two types of DHCP relay agents: global and per-interface.

- A global relay agent forwards DHCP packets to a global destination IP address.
- A per-interface relay agent is configured on a specific IP interface that is bound to a VLAN or an SPB service. Only DHCP packets originating from the VLAN or SPB service that is associated with the interface are forwarded to a destination IP address defined for the interface relay agent.

Although there are two options for configuring a DHCP Relay agent, they are mutually exclusive.Only the global DHCP Relay or per-interface DHCP Relay agent can be active at any given time. The following matrix summarizes how DHCP Request packets are processed by these two options.

Per-Interface DHCP Relay	Global DHCP Relay	Effect
Disabled	Disabled	DHCP Request is flooded within its VLAN
Disabled	Enabled	DHCP Request is relayed to the Global Relay
Enabled	Enabled	DHCP Request is relayed to the Per-Interface Relay

Configuring the Status of the DHCP Relay Feature

The global DHCP Relay or per-interface DHCP Relay configuration is not active unless the DHCP Relay feature is enabled for the switch. By default, the DHCP Relay feature is disabled. Use the **ip dhcp relay admin-state** command to change the DHCP Relay status for the switch. For example:

```
-> ip dhcp relay admin-state enable
```

When the DHCP Relay feature is enabled, DHCP packets are relayed on a global basis or on a per-interface basis. The DHCP Relay configuration for the switch determines which relay forwarding mode is active (global or per-interface).

Disabling the DHCP Relay feature does not remove the DHCP relay agent configuration from the switch.

Setting the DHCP Relay Forwarding Mode

The type of DHCP Relay agent determines the DHCP Relay forwarding mode. The global forwarding mode is active when a global DHCP Relay agent is configured; the per-interface forwarding mode is active when a DHCP Relay agent is configured for an IP interface.

Configuring the Global Relay Agent

Setting up a global DHCP Relay agent requires identifying a global destination IP address. When the global DHCP Relay mode is active, all DHCP client request are relayed to the destination IP address.

The **ip dhcp relay destination** command is used to define a global destination IP address. Specify a next hop IP address or a DHCP server IP address. For example:

```
-> ip dhcp relay destination 125.255.17.11
```

The global DHCP Relay agent forwards BOOTP/DHCP broadcasts to and from the specified address. If multiple DHCP servers are used, one IP address must be configured for each server.

To delete a global destination IP address, use the **no** form of the **ip dhcp relay destination** command. The specified IP address is deleted. For example:

```
-> no ip dhcp relay destination 125.255.17.11
```

Configuring a Relay Agent for an IP Interface

To configure a DHCP Relay agent for an IP interface, complete the following tasks:

- Identify the VLAN or SPB service to bind to an IP interface on the relay switch, and use the **ip interface** command to configure and bind the VLAN or SPB service to the IP interface. For example, the following commands create an IP interface bound to VLAN 200 and an IP interface bound to SPB service 1:

  ```
  -> ip interface guest_traffic address 10.2.2.1 vlan 200
  -> ip interface client_traffic address 2.2.0.1 service 1
  ```

  See "Configuring DHCP Relay for the SPB Service Domain" on page 23-10 for more information about configuring a relay agent for a service-bound IP interface.

- Use the **ip dhcp relay per-interface-mode** command to enable the DHCP Relay per-interface forwarding mode. For example:

  ```
  -> ip dhcp relay per-interface-mode
  ```

  Once the interface mode is enabled, global DHCP relay is no longer active.

- Use the **ip dhcp relay interface destination** command to define a destination IP address for the interface relay agent. For example, the following command configures a destination IP address to define a relay agent for the "client_traffic" interface:

  ```
  -> ip dhcp relay interface client_traffic destination 3.3.0.2
  ```

  When a destination IP address is configured for an IP interface, a relay agent is automatically enabled on that interface.

To delete a configured DHCP relay destination IP address for an IP interface, use the **no** form of the **ip dhcp relay interface destination** command. For example:

```
-> no ip dhcp relay interface client_traffic destination 3.3.0.2
```

Configuring DHCP Relay Parameters

The DHCP forward delay time and maximum hop count values are global settings that apply regardless of which DHCP Relay agent (global or per-interface) mode is active for the switch. The default values for these parameters can be accepted; changing the default values is optional.

Setting the Forward Delay

Forward Delay is a time period that gives the local server a chance to respond to a client before the relay forwards it further out in the network.

The UDP packet sent by the client contains the elapsed boot time value. This is the amount of time, measured in seconds, since the client last booted. DHCP Relay does not process the packet unless the elapsed boot time value of the client is equal to or greater than the configured value of the forward delay

time. If a packet contains an elapsed boot time value that is less than the specified forward delay time value, DHCP Relay discards the packet.

The forward delay time value applies to all defined IP DHCP Relay addresses. By default, the forward delay time is set to zero seconds. To change this value, use the **ip dhcp relay forward-delay** command. For example:

```
-> ip dhcp relay forward-delay 10
```

Setting Maximum Hops

This value specifies the maximum number of relays the BOOTP/DHCP packet can go through until it reaches its server destination. This limit keeps packets from "looping" through the network. If a UDP packet contains a hop count equal to the hops value, DHCP Relay discards the packet.

By default, the maximum number of hops defaults to 16. To change this value, use the **ip dhcp relay maximum-hops** command. For example:

```
-> ip dhcp relay maximum-hops 4
```

This maximum hops value applies only to DHCP Relay. All other switch services ignore this value.

Configuring DHCP Relay for the SPB Service Domain

As previously mentioned, DHCP Relay can be configured to relay DHCP packets between a client and a DHCP server across VLANs or over a Shortest Path Bridging (SPB) domain. Consider the following guidelines and tasks when configuring a DHCP Relay agent for the SPB service domain:

- Configure the SPB service domain network prior to attempting to configure a service-based DHCP relay agent. Refer to Chapter 7, "Configuring Shortest Path Bridging," in the *OmniSwitch AOS Release 8 Network Configuration Guide* for more information.
- Determine which SPB Backbone Edge Bridge (BEB) will relay DHCP requests between clients and the DHCP server.
- To configure a global DHCP Relay agent on the BEB relay switch, use the **ip dhcp relay destination** command to define a global destination IP address. For example, the following command configures a destination IP address to which all DHCP packets are forwarded by a global relay agent:

  ```
  -> ip dhcp relay destination 3.3.0.2
  ```

- To configure a DHCP Relay agent for an IP interface on the BEB relay switch, complete the following tasks:
  - Identify the SPB service to bind to an IP interface on the BEB relay switch and use the **ip interface** command to configure and bind the SPB service to the IP interface. For example, the following command creates the "client_traffic" interface bound to SPB service 1:

    ```
    -> ip interface client_traffic address 2.2.0.1 service 1
    ```

Notes:

- Make sure the SPB service-based IP interface is the primary IP interface for the service.
- Configuring a DHCP relay agent for an IP interface that is bound to an SPB service (a service-based IP interface) is supported only on the OmniSwitch 9900.

Refer to the "IP Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information.

– Use the **ip dhcp relay per-interface-mode** command to enable the per-interface mode. For example:

```
-> ip dhcp relay per-interface-mode
```

– Use the **ip dhcp relay interface destination** command to define a destination IP address for the interface relay agent. For example, the following command configures a destination IP address to define a relay agent for the "client_traffic" interface:

```
-> ip dhcp relay interface client_traffic destination 3.3.0.2
```

– When a destination IP address is configured for an IP interface, a relay agent is automatically enabled on that interface. To disable the relay agent, use the **ip dhcp relay interface admin-state** command. For example:

```
-> ip dhcp relay interface client_traffic admin-state disable
```

- After configuring either a global DHCP relay agent or a per-interface relay agent, use the **ip dhcp relay admin-state** command to enable the DHCP Relay for services feature on the BEB relay switch. For example:

```
-> ip dhcp relay admin-state enable
```

Sample SPB Network Topology with DHCP Relay

In the following sample SPB network topology, a DHCP relay agent is configured for a service-based IP interface on the BEB that will relay DHCP packets between the client and the server through the service domain.

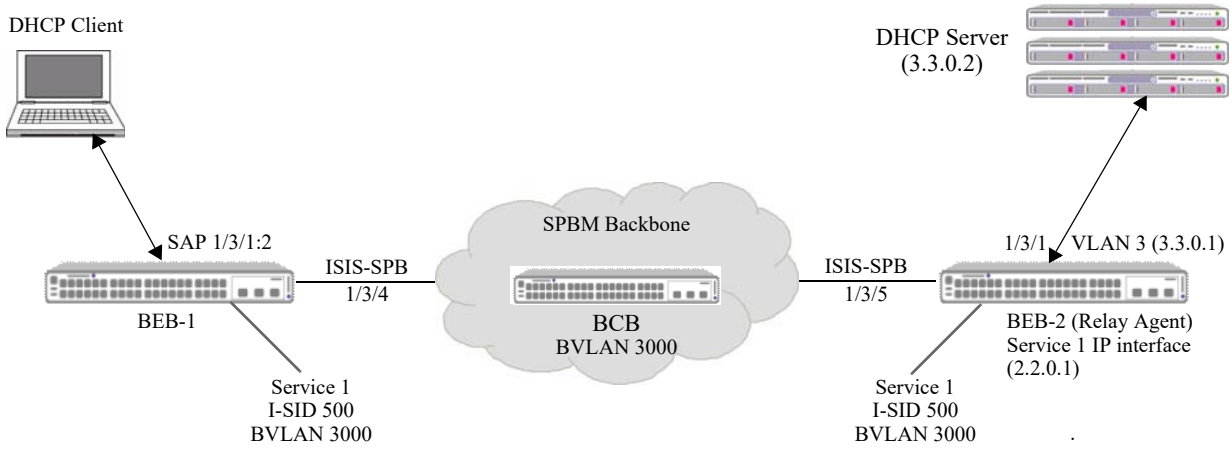


Figure 23-3 : Sample SPB Network Topology with DHCP Relay

In this sample SPB topology:

- A client sends a DHCP request for an IP address.
- The BEB-1 switch receives the client request on SAP 1/3/1:2, which is associated with SPB service 1 (I-SID 500).
- The request is then encapsulated with SPB information and sent through the SPB backbone to the BEB-2 switch on which service 1 is bound to an IP interface.
- The SPB encapsulation is then removed and the DHCP relay agent configured for the IP interface relays the packet to the DHCP server.
- The same process is followed in reverse for packets sent from the DHCP server to the client.

The sample topology is configured as follows:

- BVLAN 3000 is configured on BEB-1, BEB-2, and the BCB switch as the SPB control BVLAN.
- SPB interface ports are configured on BEB-1, BEB-2, and the BCB switch to carry traffic through the network backbone. SPB interface ports are automatically associated with BVLAN 3000.
- SPB Service 1 is configured on BEB-1 and BEB-2.
  - On BEB-1, Service 1 is associated with an SPB SAP that will process client traffic.
  - On BEB-2, Service 1 is associated with an IP interface on which the DHCP Relay Agent is configured.
- An SPB SAP (comprised of access port 1/3/1, VLAN 2 encapsulation, and associated with Service 1) is configured on BEB-1. Traffic received on port 1/3/1 tagged with VLAN 2 is encapsulated and forwarded through the SPB tunnels on which Service 1 is bound.
- On BEB-2, the “client-traffic” interface is configured with IP address 2.2.0.1 and bound to Service 1.
- A DHCP relay agent is configured for the “client-traffic” interface on BEB-2 to relay packets between the SPB service domain and the DHCP server (IP address 3.3.0.2).

The following CLI command examples are used to configure the sample SPB topology.

BEB-1:

```
-> spb bvlan 3000 admin-state enable
-> spb isis bvlan 3000 ect-id 1
-> spb isis control-bvlan 3000
-> spb isis interface port 1/3/4
-> spb isis admin-state enable
-> service 1 spb isid 500 bvlan 3000
-> service access port 1/3/1
-> service 1 sap port 1/3/1:2 admin-state enable
```

BCB:

```
-> spb bvlan 3000 admin-state enable
-> spb isis bvlan 3000 ect-id 1
-> spb isis control-bvlan 3000
-> spb isis interface port 1/3/4
-> spb isis interface port 1/3/5
-> spb isis admin-state enable
```

BEB-2:

```
-> spb bvlan 3000 admin-state enable
-> spb isis bvlan 3000 ect-id 1
-> spb isis control-bvlan 3000
-> spb isis interface port 1/3/5
-> spb isis admin-state enable
-> service 1 spb isid 500 bvlan 3000
```

BEB-2 (DHCP Relay Agent):

```
-> ip interface "client_traffic" address 2.2.0.1 mask 255.255.255.0 service 1

-> ip dhcp relay per-interface-mode
-> ip dhcp relay interface client_traffic destination 3.3.0.2

-> ip dhcp relay admin-state enable
```

Configuring the DHCP Client Interface

The OmniSwitch can be configured with a DHCP Client interface that allows the switch to obtain an IP address dynamically from a DHCP server.

- The DHCP Client interface is configurable on any one VLAN in any VRF instance.
- The DHCP Client interface supports the release and renew functionality according to RFC-2131.
- The Option-60 string can be configured on the OmniSwitch and sent as part of the DHCP discover/ request packet.
- DHCP Option-2 is supported for configuring the time zone.
- DHCP Option-12 is supported for configuring the OmniSwitch system name.

Enabling the DHCP Client Interface

To enable the DHCP Client functionality use the **ip interface dhcp-client** command. For example:

```
-> ip interface dhcp-client vlan 99
```

When the switch receives a valid IP address lease from a DHCP server:

- The IP address and the subnet mask (DHCP Option-1) are assigned to the DHCP Client IP interface.
- A default static route is created according to DHCP Option-3 (Router IP Address).
- The lease is periodically renewed and rebound according to the renew time (DHCP Option-58) and rebind time (DHCP Option-59) returned by the DHCP server. If the lease cannot be renewed within the lease time (DHCP Option-51) returned by the DHCP server, the IP address is released.
- The system name and the time zone of the OmniSwitch is set according to the system name (DHCP Option-12) and time zone (DHCP Option-2) assigned by the DHCP server. However, if the user configures the system name and time zone to non-default values, DHCP server does not assign the system name and time zone values.
- The DHCP Client-enabled IP address serves as the primary IP address when multiple addresses are configured for a VLAN.

DHCP Option-12 and DHCP Option-2

Some points to note:

- DHCP server sets the Option-2 and Option-12 values only when they are set to their default values on boot up or they are already set by the DHCP. Once the user configures these values to non-default values, DHCP does not set them.
- The user-defined configuration (through CLI, WebView, SNMP) for system name and time zone gets priority over the DHCP server values. To re-enable to DHCP mode, user has to configure the system name and time zone explicitly to their default values (if set to non-default values).

Note. User-defined values for the system name and the time zone can be set using **system name** and **system timezone** CLI commands.

- Option-2 value will be set by DHCP only if it corresponds to the timezone supported by AOS. Otherwise, this offset will be silently ignored. Refer to the *OmniSwitch AOS Release 8 Switch Management Guide* for a list of supported time zone.

Reload and Takeover

An internal file is used during a switch reload or CMM takeover to help retain the DHCP server assigned IP address. The IP address saved in this file is the address requested from the DHCP server in the event of a reload or takeover. The following information is stored in the internal file located in the ***/flash/switch*** directory on the switch:

- DHCP server assigned IP
- VLAN information
- Subnet mask
- Router IP address
- Checksum value (validates the integrity of the file).

Whenever there is any change in the DHCP server assigned IP address, the internal file is updated with the new information and synchronized to the secondary CMM. This file is also synchronized periodically with the DHCP snooping binding table.

The following occurs after a switch reload or takeover:

- The DHCP client interface uses the internal file information to create the IP interface with a lease time of 10 minutes and tries to acquire the same IP address.
- After successful renewal of the IP address, the lease time is modified as per the DHCP server assigned IP address.
- If the DHCP client is not able acquire the same IP address, the client then tries to get a new IP address after the switch-assigned DHCP lease time expires. A trap message is sent whenever there is any change to the IP address.

DHCP Client Interface Guidelines

Consider the following when configuring the DHCP Client interface:

- The IP address of a DHCP-Client interface is not configurable; this address is assigned only through the DHCP Client process of requesting an IP address.
- DHCP Client only supports IPv4 addresses.
- When using this feature in a stack configuration, enable MAC Retention to ensure that the same IP address is obtained from the DHCP server after takeover.
- Do not configure the DHCP client interface on a switch where the interface is the relay agent for the client VLAN.
- Although a DHCP Client is configurable for any VLAN in any VRF instance, only one DHCP Client per switch is allowed.
- Ensure that the DHCP server is reachable through the DHCP Client VLAN.

- When a DHCP release is performed or the DHCP client interface is deleted, any default static route added for the client is also removed and the corresponding timers (such as release/renew timer) are canceled.

DHCP Server Preference in DHCP Client Interface

DHCP server preference for the DHCP client can be configured on the switch. This allows the DHCP client to accept the lease from the highest priority server from the multiple DHCP offers received.

When server-preference is enabled in the switch. The client receives multiple DHCP OFFER messages, the server-preference logic would determine which of the DHCP servers must be given priority and the IP address provided by that DHCP server will be accepted.

The type of DHCP server sending offers will be identified by the VSI string (option 43) configuration.

When server-preference is enabled, the following precedence order is followed:

1. OVCloud: “alenterprise"
2. OVClient: “alcatel.nms.ov2500"
3. OXO: “alcatel.a4400.0"
4. Others / Undesired: Identified by absence of VSI string

A 30 second time window is activated when DHCP client interface is created with server-preference enabled. First preference is given for the OVCloud server, the client waits for 30 seconds from the time of sending discovery even after receiving OFFER from other servers other than OVCloud. If the OFFER from the OVCloud server is not received in that 30 second time period the OFFER from the next priority server or other server is accepted.

Configuring DHCP Server Preference

The DHCP server preference can be enabled using the **ip interface dhcp-client** command. For example:

```
-> ip interface dhcp-client vlan 1 server-preference
```

The server preference option can also be set without specifying VLAN ID, provided the dhcp-client interface is associated with a VLAN prior to setting the server preference. For example:

```
-> ip interface dhcp-client server-preference
```

Note. If server-preference option is enabled on a switch where dhcp-client has already obtained a lease, the obtained lease will be retained until the client triggers next DHCP discovery message.

The server-preference option is mutually exclusive with the vsi-accept-filter option. Switching from the vsi-filter setting to server-preference option is allowed only after removing the existing vsi-accept-filter option by resetting it to default value “ “. Likewise, switching from server-preference option to vsi-accept-filter setting is allowed only after removing the existing preference using '**no**' keyword.

The DHCP server preference can be removed using the **no** form of the command. For example:

```
-> ip interface dhcp-client no server-preference
```

The configured DHCP server preference option can be viewed using the **show ip interface** command.

Note. During Automatic Remote Configuration, RCL will automatically create DHCP client interface and server preference will be set to default.

Configuring DHCP Client for Preferred DHCP Server

In order to have consistency in operation, the switch should use the same IP address as originally given during RCL. The following command sets the preference DHCP server having the vendor class ID "alcatel.a4400.0" on default VLAN 1:

```
-> ip interface dhcp-client vlan 1 vsi-accept-filter "alcatel.a4400.0"
```

With vsi-asccept-filter, old lease (if any) will be released and a new lease will be obtained. To view the configured vsi-accept-filter, use the **show ip interface** command.

For more information about RCL, see the “Managing Automatic Remote Configuration Download” chapter in the *OmniSwitch AOS Release 8 Switch Management Guide*.

Configuring Generic UDP Relay

In addition to configuring a relay operation for BOOTP/DHCP traffic on the switch, it is also possible to configure relay for generic UDP service ports (DNS, NTP, other well-known UDP service ports, and service ports that are not well-known). This is done using UDP Relay commands to enable relay on these types of ports.

Note. The information presented in this section refers to both IPv4 and IPv6 UDP Relay. If there are any differences between the implementation of IPv4 and IPv6 UDP Relay, an explicit reference to IPv4 or IPv6 is made.

The generic UDP Relay function is separate from the previously described functions, such as global DHCP, per-interface DHCP, and automatic IP configuration. Using UDP Relay does not exclude or prevent the other DHCP Relay functions from working. However, the following information is important to remember when configuring BOOTP/DHCP relay and UDP port relay:

- The **ip dhcp relay** or **ipv6 dhcp relay** commands are used to configure BOOTP/DHCP (UDP ports 67/68), and the **ip udp relay** or **ipv6 udp relay** commands are used to configure generic UDP relay for other well-known and user-defined UDP ports (*not* UDP ports 67/68).
- The DHCP Relay agent relays packets between a DHCP client and a DHCP server. The next hop IP address or the DHCP server IP address is specified as the destination to which client packets are forwarded.
- The generic UDP Relay service relays packets with pre-configured destination UDP port information to destination VLANs, a Shortest Path Bridging (SPB) service, an IPv4 address, or an IPv6 address.

Configuration Guidelines

- Configuring UDP Relay for generic UDP services is a two-step process.
  - The first step involves enabling UDP Relay on the generic service port.
  - The second step involves specifying a VLAN, SPB service, IPv4 address, or IPv6 address that forwards the traffic destined for the generic service port.
- A generic UDP Relay instance can only operate at the Layer 2 or Layer 3 level at any given time.
  - When a UDP port is associated with an IPv4 or IPv6 address, the relay operates at the Layer 3 level; any attempt to then assign a VLAN or SPB service to the same UDP port is blocked.
  - When a UDP port is associated with a VLAN or SPB service, the relay operates at the Layer 2 level; any attempt to then assign an IPv4 or IPv6 address to the same UDP port is blocked.
- Broadcast UDP packets received from a client are relayed as unicast packets when the relay destination is an IPv4 or IPv6 address.
- Only one destination IPv4 or IPv6 address can be configured for a UDP port (multiple IPv4 or IPv6 addresses for the same UDP port is not supported).

Enabling/Disabling Generic UDP Relay

The following commands are used to enable an IPv4 or IPv6 UDP relay operation by specifying either a UDP service port number or a service name:

- The **ip udp relay port** or **ipv6 udp relay port** command specifies a UDP service port number to enable relay on a user-defined (not well-known) port. For example, the following commands enable generic UDP relay on UDP service port 3047:

```
-> ip udp relay port 3047
-> ipv6 udp relay port 3047
```

- The **ip udp relay service** or **ipv6 udp relay service** command specifies a well-known UDP service name instead of a UDP service port number. For example, the following commands enable relay on the DNS well-known service:

```
-> ip udp relay service dns
-> ipv6 udp relay service dns
```

To disable a relay operation for UDP service ports, use the **no** form of the above commands. For example:

```
-> ip udp relay no port 3047
-> ipv6 udp relay no port 3047

-> ip udp relay no service dns
-> ipv6 udp relay no service dns
```

For more information about using these commands, see the "DHCP Relay Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Specifying a Forwarding VLAN

The **ip udp relay vlan** or **ipv6 udp relay vlan** command is used to specify one or more VLANs on which generic UDP Relay forwards traffic destined for a UDP service port.

Note. The **ip udp relay vlan** and **ipv6 udp relay vlan** commands work only if generic UDP Relay is already enabled on the specified UDP service port.

If generic UDP Relay was enabled on a well-known UDP service port, use the **ip udp relay vlan** or **ipv6 udp relay vlan** command with the **service** parameter. For example, the following commands assign VLAN 5 as a forwarding VLAN for the DNS well-known service port:

```
-> ip udp relay service dns vlan 5
-> ipv6 udp relay service dns vlan 5
```

To specify more than one VLAN with a single command, enter a range of VLANs. For example, the following commands assign VLANs 6 through 8 as forwarding VLANs for the DNS well-known service port:

```
-> ip udp relay service dns vlan 6-8
-> ipv6 udp relay service dns vlan 6-8
```

If generic UDP Relay was enabled on a user-defined (not well-known) UDP service port, use the **ip udp relay vlan** or **ipv6 udp relay vlan** command with the **port** parameter to specify the service port number. For example, the following commands assign VLAN 100 as a forwarding VLAN for UDP service port 3047:

```
-> ip udp relay port 3047 vlan 100
-> ipv6 udp relay port 3047 vlan 100
```

To remove a VLAN association with a UDP service port, use the **no** form of the **ip udp relay vlan** or **ipv6 udp relay** command. For example:

```
-> ip udp relay service dns no vlan 6
-> ipv6 udp relay service dns no vlan 6

-> ip udp relay port 3047 no vlan 100
-> ipv6 udp relay port 3047 no vlan 100
```

For more information about using the **ip udp relay vlan** or **ipv6 udp relay vlan** command, see the "DHCP Relay Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Specifying a Forwarding SPB Service

The **ip udp relay svc** or **ipv6 udp relay svc** command is used to specify an SPB service on which generic UDP Relay forwards traffic destined for a UDP service port.

Note. The **ip udp relay svc** and **ipv6 udp relay svc** commands work only if generic UDP Relay is already enabled on the specified UDP service port.

If generic UDP Relay was enabled on a well-known UDP service port, use the **ip udp relay svc** or **ipv6 udp relay svc** command with the **service** parameter. For example, the following commands assign SPB service 10 as a forwarding service for the TFTP well-known service port:

```
-> ip udp relay service tftp svc 10
-> ipv6 udp relay service tftp svc 10
```

If generic UDP Relay was enabled on a user-defined (not well-known) UDP service port, use the **ip udp relay svc** or **ipv6 udp relay svc** command with the **port** parameter to specify the service port number. For example, the following commands assign SPB service 100 as a forwarding service for UDP service port 3047:

```
-> ip udp relay port 3047 svc 100
-> ipv6 udp relay port 3047 svc 100
```

To remove an SPB service association with a UDP service port, use the **no** form of the **ip udp relay svc** or **ipv6 udp relay svc** command. For example:

```
-> ip udp relay service tftp no svc 10
-> ipv6 udp relay service tftp no svc 10

-> ip udp relay port 3047 no svc 100
-> ipv6 udp relay port 3047 no svc 100
```

For more information about using the **ip udp relay svc** or **ipv6 udp relay svc** command, see the "DHCP Relay Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Specifying a Forwarding IP Address

The **ip udp relay address** or **ipv6 udp relay address** command is used to specify a destination IPv4 or IPv6 address to which generic UDP Relay forwards traffic destined for a UDP service port.

Note. The **ip udp relay address** and **ipv6 udp relay address** commands work only if generic UDP Relay is already enabled on the specified UDP service port.

If generic UDP Relay was enabled on a well-known UDP service port, use the **ip udp relay address** or **ipv6 udp relay address** command with the **service** parameter. For example, the following commands assign IP address 10.2.2.1 and IPv6 address 2001:db8:3001::3 as a forwarding address for the TFTP well-known service port:

```
-> ip udp relay service tftp address 10.2.2.1
-> ipv6 udp relay service tftp address 2001:db8:3001::3
```

If generic UDP Relay was enabled on a user-defined (not well-known) UDP service port, use the **ip udp relay address** or **ipv6 udp relay address** command with the **port** parameter to specify the service port number. For example, the following command assigns IP address 10.2.2.1 and IPv6 address 2001:db8:3001::3 as a forwarding address for UDP service port 3047:

```
-> ip udp relay port 3047 address 10.2.2.1
-> ipv6 udp relay port 3047 address 2001:db8:3001::3
```

To remove an IP address association with a UDP service port, use the **no** form of the **ip udp relay address** or **ipv6 udp relay address** command. For example:

```
-> ip udp relay service tftp no address 10.2.2.1
-> ipv6 udp relay service tftp no address 10.2.2.1

-> ip udp relay port 3047 no address 10.2.2.1
-> ipv6 udp relay port 3047 no address 2001:db8:3001::3
```

For more information about using the **ip udp relay address** or **ipv6 udp relay address** command, see the "DHCP Relay Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring DHCP Security Features

There are two DHCP security features available: DHCP relay agent information option (Option-82) and DHCP Snooping.

- The DHCP Option-82 feature enables the relay agent to insert identifying information into client-originated DHCP packets before the packets are forwarded to the DHCP server.
- The DHCP Snooping feature filters DHCP packets between untrusted sources and a trusted DHCP server and builds a binding database to log DHCP client information.

Although DHCP Option-82 is a subcomponent of DHCP Snooping, these two features are mutually exclusive. If the DHCP Option-82 feature is enabled for the switch, then DHCP Snooping is not available. The reverse is also true; if DHCP Snooping is enabled, then DHCP Option-82 is not available. In addition, the following differences exist between these two features:

- DHCP Snooping does require and use the Option-82 data insertion capability, but does not implement any other behaviors defined in RFC 3046.
- DHCP Snooping is configurable at the switch level and on a per-VLAN basis, but DHCP Option-82 is only configurable at the switch level.

Note. DHCP Snooping provides multiple VLAN tagging support.

The following sections provide additional information about each DHCP security feature and how to configure feature parameters using the Command Line Interface (CLI).

Using the Relay Agent Information Option (Option-82)

This implementation of the DHCP relay agent information option (Option-82) feature is based on the functionality defined in RFC 3046. By default DHCP Option-82 functionality is disabled. The **ip dhcp relay insert-agent-information** command is used to enable this feature at the switch level.

When this feature is enabled, communications between a DHCP client and a DHCP server are authenticated by the relay agent. To accomplish this task, the agent adds Option-82 data to the end of the options field in DHCP packets sent from a client to a DHCP server. Option-82 consists of two suboptions: Circuit ID and Remote ID. The agent fills in the following information for each of these suboptions:

- **Circuit ID**—the VLAN ID and slot/port from where the DHCP packet originated.
- **Remote ID**—the MAC address of the router interface associated with the VLAN ID specified in the Circuit ID suboption.

The **ip dhcp relay insert-agent-information format** command is used to configure the type of data (base MAC address, system name, interface alias, or user-defined) that is inserted into the above Option-82 suboptions. The system name and user-defined text are reported in ASCII text format, but the MAC address is still reported in hex-based format.

By default, the relay agent drops client DHCP packets it receives that already contain Option-82 data. However, it is possible to configure an Option-82 policy to specify how such packets are treated. See "Configuring a Relay Agent Information Option-82 Policy" on page 23-23 for more information.

The DHCP Option-82 feature is only applicable when DHCP relay is used to forward DHCP packets between clients and servers associated with different VLANs. In addition, a secure IP network must exist between the relay agent and the DHCP server.

How the Relay Agent Processes DHCP Packets from the Client

The following table describes how the relay agent processes DHCP packets received from clients when the Option-82 feature is enabled for the switch:

If the DHCP packet from the client ...	The relay agent ...
Contains a zero gateway IP address (0.0.0.0) and no Option-82 data.	Inserts Option-82 with unique information to identify the client source.
Contains a zero gateway IP address (0.0.0.0) and Option-82 data.	Drops the packet, keeps the Option-82 data and forwards the packet, or replaces the Option-82 data with its own Option-82 data and forwards the packet. The action performed by the relay agent in this case is determined by the agent information policy that is configured through the **ip dhcp relay insert-agent-information policy** command. By default, this type of DHCP packet is dropped by the agent.
Contains a non-zero gateway IP address and no Option-82 data.	Drops the packet without any further processing.
Contains a non-zero gateway IP address and Option-82 data.	Drops the packet if the gateway IP address matches a local subnet, otherwise the packet is forwarded without inserting Option-82 data.

How the Relay Agent Processes DHCP Packets from the Server

When the relay agent receives a DHCP packet from the DHCP server, the agent:

1 Extracts the VLAN ID from the Circuit ID suboption field in the packet and compares the MAC address of the IP router interface for that VLAN to the MAC address contained in the Remote ID suboption field in the same packet.

2 Drops the DHCP packet if the IP router interface MAC address and the Remote ID MAC address are not the same.

3 If the two MAC addresses match, then a check is made to see if the slot/port value in the Circuit ID suboption field in the packet matches a port that is associated with the VLAN also identified in the Circuit ID suboption field.

4 If the slot/port information does not identify an actual port associated with the Circuit ID VLAN, then the agent tries to deliver the packet back to the port where the device is located.

5 If the slot/port information does identify an actual port associated with the Circuit ID VLAN, then the agent strips the Option-82 data from the packet and unicasts the packet to the port identified in the Circuit ID suboption.

Enabling the Relay Agent Information Option-82

Use the **ip dhcp relay insert-agent-information** command to enable the DHCP Option-82 feature for the switch. For example:

```
-> ip dhcp relay insert-agent-information
```

Use the **no** form of this command to disable the DHCP Option-82 feature. For example:

```
-> no ip dhcp relay insert-agent-information
```

DHCP Option-82 functionality is not restricted to ports associated with a specific VLAN as this feature is not available on a per-VLAN basis. Instead, DHCP traffic received on all ports is eligible for Option-82 data insertion when it is relayed by the agent.

Configuring a Relay Agent Information Option-82 Policy

As previously mentioned, when the relay agent receives a DHCP packet from a client that already contains Option-82 data, the packet is dropped by default. However, it is possible to configure a DHCP Option-82 policy that directs the relay agent to drop, keep, or replace the existing Option-82 data and then forward the packet to the server.

To configure a DHCP Option-82 policy, use the **ip dhcp relay insert-agent-information policy** command. The following parameters are available with this command to specify the policy action:

- **drop**—The DHCP Option-82 data is dropped (the default).
- **keep**—The existing Option-82 field information in the DHCP packet is retained and the packet is relayed to the DHCP server.
- **replace**—The existing Option-82 data in the DHCP packet is replaced with the VLAN ID and the MAC address of the DHCP Relay switch.

For example, the following commands configure DHCP Option-82 policies:

```
-> ip dhcp relay insert-agent-information policy drop
-> ip dhcp relay insert-agent-information policy keep
-> ip dhcp relay insert-agent-information policy replace
```

Note. These policies apply to all DHCP packets received on all switch ports. In addition, if a packet that contains existing Option-82 data also contains a gateway IP address that matches a local subnet address, the relay agent drops the packet.

Using DHCP Snooping

Using DHCP Snooping improves network security by filtering DHCP messages received from devices outside the network and building and maintaining a binding table (database) to track access information for such devices.

In order to identify DHCP traffic that originates from outside the network, DHCP Snooping categorizes ports as either trusted or untrusted. A port is trusted if it is connected to a device inside the network, such as a DHCP server. A port is untrusted if it is connected to a device outside the network, such as a customer switch or workstation.

Additional DHCP Snooping functionality provided includes the following:

- **Layer 2 DHCP Snooping—**Applies DHCP Snooping functionality to bridged DHCP client/server broadcasts without using the relay agent or requiring an IP interface on the client/server VLAN. See "Layer 2 DHCP Snooping" on page 23-26 for more information.
- **IP Source Filtering (Dynamic ARP Inspection - (DAI))** —Restricts DHCP Snooping port traffic to only packets that contain the proper client source information. The DHCP Snooping binding table is used to verify the client information for the port that is enabled for IP source filtering. See "Configuring IP Source Filtering (Dynamic ARP Inspection (DAI))" on page 23-28 for more information.
- **Rate Limiting**—Limits the rate of DHCP packets on the port. This functionality is achieved using the QoS application to configure ACLs for the port. See Chapter 28, "Configuring QoS," in the *OmniSwitch AOS Release 8 Network Configuration Guide* for more information.

When DHCP Snooping is first enabled, all ports are considered untrusted. It is important to then configure ports connected to a DHCP server inside the network as trusted ports. See "Configuring the Port Trust Mode" on page 23-27 for more information.

If a DHCP packet is received on an untrusted port, then it is considered an untrusted packet. If a DHCP packet is received on a trusted port, then it is considered a trusted packet. DHCP Snooping only filters untrusted packets and will drop such packets if one or more of the following conditions are true:

- The packet received is a DHCP server packet, such as a DHCPOFFER, DHCPACK, or DHCPNAK packet. When a server packet is received on an untrusted port, DHCP Snooping knows that it is not from a trusted server and discards the packet.
- The source MAC address of the packet and the DHCP client hardware address contained in the packet are not the same address.
- The packet is a DHCPRELEASE or DHCPDECLINE broadcast message that contains a source MAC address found in the DHCP Snooping binding table, but the interface information in the binding table does not match the interface on which the message was received.
- The packet includes a relay agent IP address that is a non-zero value.
- The packet already contains Option-82 data in the options field and the Option-82 check function is enabled. See "Bypassing the Option-82 Check on Untrusted Ports" on page 23-27 for more information.

If none of the above are true, then DHCP Snooping accepts and forwards the packet. When a DHCPACK packet is received from a server, the following information is extracted from the packet to create an entry in the DHCP Snooping binding table:

- MAC address of the DHCP client.

- IP address for the client that was assigned by the DHCP server.
- The port from where the DHCP packet originated.
- The VLAN associated with the port from where the DHCP packet originated.
- The lease time for the assigned IP address.
- The binding entry type; dynamic or static (user-configured).

After extracting the above information and populating the binding table, the packet is then forwarded to the port from where the packet originated. Basically, the DHCP Snooping feature prevents the normal flooding of DHCP traffic. Instead, packets are delivered only to the appropriate client and server ports.

DHCP Snooping Configuration Guidelines

Consider the following when configuring the DHCP Snooping feature:

- Layer 3 DHCP Snooping requires the use of the relay agent to process DHCP packets. As a result, DHCP clients and servers must reside in different VLANs so that the relay agent is engaged to forward packets between the VLAN domains. See "Configuring DHCP Relay" on page 23-8 for information about how to configure the relay agent on the switch.
- Layer 2 DHCP Snooping does not require the use of the relay agent to process DHCP packets. As a result, an IP interface is not needed for the client/server VLAN. See "Layer 2 DHCP Snooping" on page 23-26 for more information.
- Both Layer 2 and Layer 3 DHCP Snooping are active when DHCP Snooping is globally enabled for the switch or enabled on a one or more VLANs. See "Enabling DHCP Snooping" on page 23-25 for more information.
- Configure ports connected to DHCP servers within the network as trusted ports. See "Configuring the Port Trust Mode" on page 23-27 for more information.
- Make sure that Option-82 data insertion is always enabled at the switch or VLAN level. See "Enabling DHCP Snooping" on page 23-25 for more information.
- DHCP packets received on untrusted ports that already contain the Option-82 data field are discarded by default. To accept such packets, configure DHCP Snooping to bypass the Option-82 check. See "Bypassing the Option-82 Check on Untrusted Ports" on page 23-27 for more information.

Enabling DHCP Snooping

There are two levels of operation available for the DHCP Snooping feature: switch level or VLAN level. These two levels are exclusive of each other in that they both cannot operate on the switch at the same time. In addition, if the global DHCP relay agent information option (Option-82) is enabled for the switch, then DHCP Snooping at any level is not available. See "Using the Relay Agent Information Option (Option-82)" on page 23-21 for more information.

Note. DHCP Snooping drops server packets received on untrusted ports (ports that connect to devices outside the network or firewall). It is important to configure ports connected to DHCP servers as trusted ports so that traffic to/from the server is not dropped.

Switch-level DHCP Snooping

By default, DHCP Snooping is disabled for the switch. To enable this feature at the switch level, use the **dhcp-snooping admin-state** command. For example:

```
-> dhcp-snooping admin-state enable
```

When DHCP Snooping is enabled at the switch level, all DHCP packets received on all switch ports are screened/filtered by DHCP Snooping. By default, only client DHCP traffic is allowed on the ports, unless the trust mode for a port is configured to block or allow all DHCP traffic. See "Configuring the Port Trust Mode" on page 23-27 for more information.

In addition, the following functionality is also activated by default when switch-level DHCP Snooping is enabled:

- The DHCP Snooping binding table is created and maintained. To configure the status, use the **dhcp-snooping binding admin-state** command; to add a static entry to this table, use the **dhcp-snooping binding** command.
- MAC address verification is performed to compare the source MAC address of the DHCP packet with the client hardware address contained in the packet. To configure the status of MAC address verification, use the **dhcp-snooping mac-address-verification** command.
- Option-82 data is inserted into the packet and then DHCP reply packets are only sent to the port from where the DHCP request originated, instead of flooding these packets to all ports. To configure the status of Option-82 data insertion, use the **dhcp-snooping option-82-data-insertion** command.
- The base MAC address of the switch is inserted into the Circuit ID and Remote ID suboptions of the Option-82 field. To configure the type of data (base MAC address, system name, or user-defined) that is inserted into the Option-82 suboptions, use the **dhcp-snooping option-82 format** command. The system name and user-defined text are reported in ASCII text format, but the MAC address is still reported in hex-based format.

VLAN-Level DHCP Snooping

To enable DHCP Snooping at the VLAN level, use the **dhcp-snooping vlan** command. For example, the following command enables DHCP Snooping for VLAN 200:

```
-> dhcp-snooping vlan 200 admin-state enable
```

When this feature is enabled at the VLAN level, DHCP Snooping functionality is only applied to ports that are associated with a VLAN that has this feature enabled.

Note. Enabling DHCP Snooping at the switch level is not allowed if it is enabled for one or more VLANs.

By default, when DHCP Snooping is enabled for a specific VLAN, MAC address verification and Option-82 data insertion is also enabled for the VLAN by default. To disable or enable either of these two features, use the **dhcp-snooping vlan** command with either the **mac-address-verification** or **option-82-data-insertion** parameters. For example:

```
-> dhcp-snooping vlan 200 mac-address-verification disable
-> dhcp-snooping vlan 200 option-82-data-insertion disable
```

Note. If DHCP Snooping is *not* enabled for a VLAN, then all ports associated with the VLAN are considered trusted ports. VLAN-level DHCP Snooping does not filter DHCP traffic on ports associated with a VLAN that does not have this feature enabled.

Layer 2 DHCP Snooping

By default, DHCP broadcasts are flooded on the default VLAN of the client or server port. If the DHCP client and server are both members of the same VLAN domain, the broadcast packets from these sources are bridged as Layer 2 traffic and not processed by the relay agent.

When DHCP Snooping is enabled at the switch level or for an individual VLAN, DHCP Snooping functionality is also applied to Layer 2 traffic. When DHCP Snooping is disabled at the switch level or disabled on the last VLAN to have snooping enabled on the switch, DHCP Snooping functionality is no longer applied to Layer 2 or Layer 3 traffic.

Configuring the Port Trust Mode

The DHCP Snooping trust mode for a port determines whether or not the port accepts all DHCP traffic, client-only DHCP traffic, or blocks all DHCP traffic. The following trust modes for a port are configurable using the **dhcp-snooping port** command:

- **client-only**—The default mode applied to ports when DHCP Snooping is enabled. This mode restricts DHCP traffic on the port to only DHCP client-related traffic. When this mode is active for the port, the port is considered an untrusted interface.
- **trust**—This mode does not restrict DHCP traffic on the port. When this mode is active on a port, the port is considered a trusted interface. In this mode the port behaves as if DHCP Snooping is not enabled.
- **block**—This mode blocks all DHCP traffic on the port. When this mode is active for the port, the port is considered an untrusted interface.

To configure the trust mode for one or more ports, use the **dhcp-snooping port** command. For example, the following command changes the trust mode for port 1/12 on chassis 1 to blocked:

```
-> dhcp-snooping port 1/1/12 block
```

It is also possible to specify a range of ports. For example, the following command changes the trust mode for ports 2/1 through 2/10 on chassis 1 to trusted:

```
-> dhcp-snooping port 1/2/1-10 trust
```

Note. It is necessary to configure ports connected to DHCP servers within the network and/or firewall as trusted ports so that necessary DHCP traffic to/from the server is not blocked. Configuring the port mode as trusted also identifies the device connected to that port as a trusted device within the network.

Bypassing the Option-82 Check on Untrusted Ports

By default, DHCP Snooping checks packets received on untrusted ports (DHCP Snooping client-only or blocked ports) to see if the packets contain the Option-82 data field. If a packet does contain this field, the packet is dropped.

To allow untrusted ports to receive and process DHCP packets that already contain the Option-82 data field, use the **dhcp-snooping bypass option-82-check** command to disable the Option-82 check. For example:

```
-> dhcp-snooping bypass option-82-check enable
```

Configuring the Option-82 Policy

The Option-82 policy specifies whether to keep, replace, or drop the Option-82 field from DHCP packets entering the switch.

By default the Option-82 policy replaces the Option-82 filed in the incoming DHCP packets. To configure the Option-82 policy for the incoming DHCP packets, use the **dhcp-snooping option-82 policy** command. For example, the following configuration allows to keep the Option-82 field in the DHCP packet:

```
-> dhcp-snooping option-82 policy keep
```

Configuring IP Source Filtering (Dynamic ARP Inspection (DAI))

IP source filtering applies to DHCP Snooping ports and restricts port traffic to only packets that contain the proper client source information in the packet. The DHCP Snooping binding table is used to verify the client information for the port that is enabled for IP source filtering.

Port Source Filtering—Filters based on source MAC address and source IP address. DHCP Snooping IP Source Filtering commands also support port ranges to reduce the number of lines of configuration.

VLAN Source Filtering—Filters based on VLAN ID, interface number, source MAC address and source IP address.

Configuring the Global IP Source Filtering Status

The IP source filtering configuration for a port or VLAN is not active unless the IP source filtering feature is globally enabled for the switch. By default, the IP source filtering functionality is enabled for the switch. To change the global IP source filtering status, use the **dhcp-snooping ip-source-filter admin-state** command. For example:

```
-> dhcp-snooping ip-source-filter admin-state disable
```

Globally disabling the IP source filtering functionality does not remove the user-defined IP source configuration from the switch. The existing port and VLAN source filtering configuration is maintained but is not operationally active.

In addition, the global IP source filtering status is not changed when the DHCP Snooping status is changed. For example, if DHCP Snooping is disabled for the switch and IP source filtering is enabled, IP source filtering functionality is still enabled and applied to static binding table entries.

Configuring Port and VLAN Source Filtering

By default IP source filtering is disabled for a DHCP Snooping port, link aggregate, or VLAN. Use the **dhcp-snooping ip-source-filter** command to enable or disable this function.

To enable IP source filtering on a port, use the **dhcp-snooping ip-source-filter** command with the **port** option. For example:

```
-> dhcp-snooping ip-source-filter port 1/1/1 admin-state enable
```

To enable source filtering on link aggregate, use the **dhcp-snooping ip-source-filter** command with the **linkagg** option. For example:

```
-> dhcp-snooping ip-source-filter linkagg 2 admin-state enable
```

To enable source filtering on a VLAN, use the **dhcp-snooping ip-source-filter** command with the **vlan** option. For example:

```
-> dhcp-snooping ip-source-filter vlan 10 admin-state enable
```

Configuring the DHCP Snooping Binding Table

The DHCP Snooping binding table is automatically enabled by default when DHCP Snooping is enabled at either the switch or VLAN level. This table is used by DHCP Snooping to filter DHCP traffic that is received on untrusted ports.

Entries are made in this table when the relay agent receives a DHCPACK packet from a trusted DHCP server. The agent extracts the client information, populates the binding table with the information and then forwards the DHCPACK packet to the port where the client request originated.

To enable or disable the DHCP Snooping binding table, use the **dhcp-snooping binding admin-state** command. For example:

```
-> dhcp-snooping binding admin-state enable
-> dhcp-snooping binding admin-state disable
```

Note that enabling the binding table functionality is not allowed if Option-82 data insertion is *not* enabled at either the switch or VLAN level.

In addition, it is also possible to configure static binding table entries. This type of entry is created using the **dhcp-snooping binding** command parameters to define the static entry. For example, the following command creates a static DHCP client entry:

```
-> dhcp-snooping binding 00:2a:95:51:6c:10 port 1/1/15 address 17.15.3.10 vlan
200
```

To remove a static binding table entry, use the **no** form of the **dhcp-snooping binding** command. For example:

```
-> no dhcp-snooping binding 00:2a:95:51:6c:10 port 1/1/15 address 17.15.3.10
vlan 200
```

To view the DHCP Snooping binding table contents, use the **show dhcp-snooping binding** command.

DHCP snooping binding entries can be displayed in a static or dynamic snapshot format, so that the content can be copied-pasted for reconfiguration when in need.

To filter DHCP snooping static binding entries, use the command as follows:

```
-> show dhcp-snooping binding snapshot static
dhcp-snooping binding 11:22:33:44:00:00 port 1/1 address 1.1.1.1 vlan 10
dhcp-snooping binding 11:22:33:44:11:11 port 1/2 address 1.1.1.2 vlan 10
dhcp-snooping binding 11:22:33:44:22:22 port 1/3 address 1.1.1.3 vlan 10dhcp-snoop-
ing binding 11:22:33:44:33:33 port 1/4 address 1.1.1.4 vlan 10
```

To filter DHCP snooping dynamic binding entries, use the command as follows;

```
-> show dhcp-snooping binding snapshot dynamic
dhcp-snooping binding 11:22:33:44:00:44 port 1/1 address 1.1.1.11 vlan 10
dhcp-snooping binding 11:22:33:44:11:55 port 1/2 address 1.1.1.12 vlan 10
dhcp-snooping binding 11:22:33:44:22:66 port 1/3 address 1.1.1.13 vlan 10dhcp-
snooping binding 11:22:33:44:33:77 port 1/4 address 1.1.1.14 vlan 10
```

For more information, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring the Binding Table Timeout

The contents of the DHCP Snooping binding table resides in the switch memory. In order to preserve table entries across switch reboots, the table contents is automatically saved to the **dhcpBinding.db** file located in the **/flash/switch** directory.

Note. Do not manually change the **dhcpBinding.db** file. This file is used by DHCP Snooping to preserve and maintain binding table entries. Changing the file name or contents can cause problems with this functionality or with the DHCP Snooping application itself.

The amount of time, in seconds, between each automatic save is referred to as the binding table timeout value. By default, the timeout value is 1 second. To configure this value, use the **dhcp-snooping binding timeout** command. For example, the following command sets the timeout value to 300 seconds:

```
-> dhcp-snooping binding timeout 300
```

Each time an automatic save is performed, the **dhcpBinding.db** file is time stamped.

Synchronizing the Binding Table

To synchronize the contents of the **dhcpBinding.db** file with the binding table contents that resides in memory, use the **dhcp-snooping binding action** command. This command provides three parameters: **purge**, **renew**, and **save**.

- Use the **purge** parameter to clear binding table entries in memory.
- Use the **renew** parameter to populate the binding table with the contents of the **dhcpBinding.db** file.
- Use the **save** parameter to explicitly save of the binding table entries in memory to the **dhcpBinding.db** file.

For example:

```
-> dhcp-snooping binding action purge
-> dhcp-snooping binding action renew
-> dhcp-snooping binding action save
```

Synchronizing the binding table is only done when this command is used. There is no automatic triggering of this function. In addition, it is important to note that synchronizing the binding table loads **dhcpBinding.db** file contents into memory. This is the reverse of saving the binding table contents in memory to the **dhcpBinding.db** file, which is done at automatic time intervals as defined by the binding table timeout value. See "Configuring the Binding Table Timeout" on page 23-29 for more information.

Binding Table Retention

When the binding table is synchronized with the contents of the **dhcpBinding.db** file, any table entries with a MAC address that no longer appears in the MAC address table are cleared from the binding table. To retain these entries regardless of their MAC address table status, use the **dhcp-snooping binding persistency** command. For example:

```
-> dhcp-snooping binding persistency enable
```

When binding table retention is enabled, entries remain in the table for the term of their DHCP lease and are not removed even when the MAC address for the entry is cleared from the MAC address table. However, when the DHCP client host is connected to another port, the associated binding table is moved from the previous port to the new port.

To disable binding table retention, use the following command:

```
-> dhcp-snooping binding persistency disable
```

Verifying the DHCP Relay Configuration

To display information about the DHCP Relay and BOOTP/DHCP, use the **show** commands listed below.

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. An example of the output for the **show ip dhcp relay interface** command is also given in "Quick Steps for Setting Up DHCP Relay" on page 23-3.

show ip dhcp relay interface	Displays the current forward delay time, the maximum number of hops, the forwarding option (standard), and each of the DHCP server IP addresses configured.
show ip dhcp relay statistics	Displays the number of packets the DHCP Relay service has received and transmitted, the number of packets dropped due to forward delay and maximum hops violations, and the number of packets processed since the last time these statistics were displayed.
show ip udp relay	Displays the VLAN assignments to which the traffic received on the UDP service ports is forwarded. Displays the current configuration for UDP services by service name or by service port number.
show ip udp relay statistics	Displays the current statistics for each UDP port relay service. These statistics include the name of the service, the forwarding VLAN(s) configured for that service, and the number of packets the service has sent and received.
show dhcp-snooping vlan	Displays a list of VLANs that have DHCP Snooping enabled and whether or not MAC address verification and Option-82 data insertion is enabled for each VLAN.
show dhcp-snooping port	Displays the DHCP Snooping trust mode for the port and the number of packets destined for the port that were dropped due to a DHCP Snooping violation.
show dhcp-snooping binding	Displays the contents of the DHCP Snooping binding table (database).

Notes.

- Use the **ip dhcp relay clear statistics** command to reset the DHCP Relay statistics for VRF instances.
- Use the **ip udp relay no statistics** command to reset the generic UDP Relay Service related statistics.

DHCPv6 Relay Overview

A DHCPv6 relay agent is required in situations where DHCPv6 clients do not reside on the same link as the DHCP server. When an OmniSwitch is acting as a DHCPv6 Relay Agent, it relays all the messages from a DHCPv6 client to one or more relay destinations. Relay destinations may be a DHCPv6 server or another relay agent. Replies from a DHCPv6 server are relayed back to the client.

DHCPv6 Relay is a per-interface option that can be enabled on any IPv6 interface except loopback, loopback0, or Ethernet Management Port (EMP).

For details on how DHCPv6 Relay and configuration is implemented on the OmniSwitch, see the following sections.

Quick Steps for Configuring DHCPv6 Relay

To configure the DHCPv6 relay feature, proceed as follows:

1 Enable the DHCPv6 relay service on the switch using the **ipv6 dhcp relay admin-state** command. For example:

```
-> ipv6 dhcp relay admin-state enable
```

Note. Though the DHCPv6 relay is enabled on the switch, DHCPv6 relay has to be explicitly enabled on the interface for the DHCPv6 client messages to be relayed. At least one relay destination must be configured before enabling the DHCPv6 relay on an interface.

2 Configure the DHCPv6 relay destination using the **ipv6 dhcp relay destination** command. For example, the following command configures a relay destination address for the “int1” interface:

```
-> ipv6 dhcp relay int1 destination 2001:DB8:3001::3
```

Note. If the relay destination is a link-local, then the interface name for the link-local must be specified. For example:

```
-> ipv6 dhcp relay v6vlan10 destination fe80::64 v6vlan20
```

3 Configure an interface to relay DHCPv6 client messages using the **ipv6 dhcp relay interface admin-state** command. For example, the following command enables DHCPv6 relay on the “int1” interface.

```
-> ipv6 dhcp relay int1 admin-state enable
```

Note. At least one relay destination must be configured before enabling the DHCPv6 relay on an interface.

DHCPv6 Relay Interface

The DHCPv6 relay can operate in multiple VRFs (IPv6 multi-VRF). It supports multicast-capable IPv6 interfaces (VLAN, service interfaces, and configured tunnel interfaces) and non-multicast-capable IPv6 interfaces (6to4 tunnel).

The DHCPv6 Relay agent will be part of the link-scoped multicast group (FF02::1:2) on the interface. Any messages sent by a client to that address will then be handled by DHCPv6 Relay agent.

A maximum of five unicast or link-scoped multicast relay destinations can be configured for each interface on which DHCPv6 Relay is enabled. The DHCPv6 relay for the interface will be automatically disabled when all the relay destinations configured for that interface is removed.

DHCPv6 Relay Messages

Relay-Forward Messages

A Relay-Forward message is used to relay a client message to the DHCPv6 server (or another relay agent).

There can be multiple relay agents between a DHCPv6 client and the server. When the DHCPv6 relay agent receives the relay-forward message, it will verify the hop-count of the message. If the hop-count value is greater or equal to the hop-count limit (32), the message will be discarded and a new relay-forward message is created else the message is forwarded to the relay destination.

Relay-Reply Messages

When a DHCPv6 Server responds to a relayed DHCPv6 client message the response is sent in a Relay-Reply message.

When a relay agent receives a Relay-Reply message, it will extract the encapsulated message and relay it to the peer address specified in the Relay-Reply. The peer address may be for the client or another relay agent. If the peer address is not reachable, the reply message will be discarded.

Configuring DHCPv6 Relay

The following sections provide the details about defining a DHCPv6 Relay configuration.

Enabling the DHCPv6 Relay Service

The DHCPv6 relay service can be enabled on a per-VRF basis, use the **ipv6 dhcp relay admin-state** command. For example:

```
-> ipv6 dhcp relay admin-state enable
-> vrf vrf1
vrf1::-> ipv6 dhcp relay admin-state enable
```

Note. Though the DHCPv6 relay is enabled on the switch, it has to be explicitly enabled on the interface for the DHCPv6 client messages to be relayed.

Configuring the DHCPv6 Relay Interface

DHCPv6 relay must be configured on an IPv6 interface in order to relay DHCPv6 messages between clients and the server. To configure a DHCP Relay agent for an IPv6 interface, complete the following tasks:

1 Identify the VLAN or SPB service to bind to an IPv6 interface on the relay switch, and use the **ipv6 interface** command to configure and bind the VLAN or SPB service to the IPv6 interface. For example, the following commands create an IPv6 interface bound to VLAN 10 and an IPv6 interface bound to SPB service 1:

```
-> ipv6 interface v6vlan10 vlan 10
-> ipv6 interface v6vlan10 service 1
```

See "Configuring DHCP Relay for the SPB Service Domain" on page 23-10 for more information about configuring a relay agent for a service-bound IP interface.

2 Use the **ipv6 dhcp relay destination** command to configure a destination IPv6 address (a DHCPv6 Server or another relay agent) for the interface relay agent. For example, the following command configures a destination IPv6 address to define a relay agent for the "v6vlan10" interface:

```
-> ipv6 dhcp relay v6vlan10 destination 2001:db8:3001::3
```

If the destination address is a link-local address, then the name of the interface used to reach the destination must also be specified. For example:

```
-> ipv6 dhcp relay v6vlan10 destination fe80::99 v6vlan20
```

Notes:

- Packets destined for the specified IPv6 address are relayed over the VLAN or SPB service domain.
- Configuring a DHCPv6 relay agent for an IPv6 interface that is bound to an SPB service (a service-based IPv6 interface) is supported only on the OmniSwitch 9900.

3 Use the **ipv6 dhcp relay interface admin-state** command to enabled DHCPv6 relay on the interface. For example:

```
-> ipv6 dhcp relay v6vlan10 admin-state enable
```

Note. If all relay destinations configured for an interface are removed, DHCPv6 relay on the interface will be automatically disabled.

Setting Maximum Hops

This value specifies the maximum number of relays the DHCPv6 packet can go through until it reaches its server destination. This limit keeps packets from "looping" through the network. If a UDP packet contains a hop count equal to the hops value, DHCPv6 Relay discards the packet.

By default, the maximum number of hops defaults to 32. To change this value, use the **ipv6 dhcp relay maximum-hops** command. For example:

```
-> ipv6 dhcp relay maximum-hops 10
```

This maximum hops value applies only to DHCPv6 Relay. All other switch services ignore this value.

Verifying the DHCPv6 Relay Configuration

The interface for which the DHCPv6 relay is configured, the relay destinations, and the status of the DHCPv6 relay can be displayed. To view the DHCPv6 Relay configuration, use the **show ipv6 dhcp relay** command. For example:

```
-> show ipv6 dhcp relay
DHCPv6 Relay: Enabled
Maximum Hops: 32

Interface                   Relay Destination(s)                      Status
---------------------------+----------------------------------------+--------
vlan-41                     ff02::1:2                                 Enabled
vlan-103                    2001:dbc8:8003::17                        Disabled
                            2001:dbc8:8004::99
vlan-200                    fe80::cd0:deff:fe28:1ca5 vlan-201         Enabled
tunnel-2                    2001:dbc8:a23::ea77                       Enabled
```

Using DHCPv6 Snooping

DHCPv6 Snooping monitors DHCPv6 client and server exchanges passing through the switch. It builds a binding table database of DHCPv6-assigned addresses based on the contents of those exchanges. The binding table is used by IPv6 Source Filtering to prevent unauthorized hosts from sending packets via the switch.

Enabling DHCPv6 Snooping

DHCPv6 Snooping can be enabled globally or per-VLAN basis. The global DHCPv6 Snooping and per-VLAN DHCPv6 Snooping are mutually exclusive.

Note. DHCPv6 Snooping must not be used in configurations where a DHCPv6 server assigns multiple addresses to a client.

Per-VLAN DHCPv6 Snooping

To enable DHCPv6 Snooping at the VLAN level, use the **dhcpv6-snooping vlan admin-state** command. For example, the following command enables DHCPv6 Snooping for VLAN 200:

```
-> dhcpv6-snooping vlan 200 admin-state enable
```

When enabled on a VLAN, DHCPv6 Snooping will monitor all DHCPv6 client and server exchanges and populate the binding table based on the message contents.

The global DHCPv6 Snooping must be disabled before enabling the per-VLAN DHCPv6 Snooping.

Global DHCPv6 Snooping

Apart from VLAN-Level DHCPv6 Snooping, DHCPv6 Snooping can be enabled globally for the switch. To enable this feature globally, use the **dhcpv6-snooping global admin-state** command. For example:

```
-> dhcpv6-snooping global admin-state enable
```

The per-VLAN DHCPv6 Snooping must be disabled before enabling global DHCPv6 Snooping. When global DHCPv6 Snooping is enabled, the DHCPv6 Snooping binding table will be constructed based on DHCPv6 client and server exchanges seen on any VLAN.

Configuring the DHCPv6 Snooping Binding Table

There are two types of binding entries:

- Static binding entry: These entries are created using the CLI. The entries are stored in the configuration file and the configuration must be manually removed.
- Dynamic binding entry: These entries are created by learning the DHCPv6 packet exchange between the client and the DHCPv6 server when snooping is enabled.The entries are stored in a permanent file in the memory and is retrieved upon takeover or reboot. These binding entries will get deleted upon lease time expiry and also during link down or MAC address deletion unless persistency is enabled on the switch.

Configuring the Binding Table Timeout

The DHCPv6 Snooping binding table is saved to permanent storage so that the information contained in it is preserved across switch reboots.

By default, the binding table is saved to permanent storage at one second interval (if any changes have occurred). The save interval can be modified using the **dhcpv6-snooping binding timeout** command. For example, to change the save interval to 50 seconds:

```
-> dhcpv6-snooping binding timeout 50
```

Binding Table Actions

The following actions can be performed on the binding table:

- **Purge** allows to immediately flush the contents of the binding table. For example:

```
-> dhcpv6-snooping binding action purge
```

- **Renew** allows to reload the binding table from that most recently saved to permanent storage. For example:

```
-> dhcpv6-snooping binding action renew
```

- **Save** allows to trigger an immediate save of the binding table contents to permanent storage, regardless of any binding table save timeout configuration. For example:

```
-> dhcpv6-snooping binding action save
```

Note. While using binding table action commands, ensure the binding timeout interval is set greater than 10 seconds from the default interval 1 second to avoid quick timeout.

Binding Table Retention

When the binding table is synchronized, the dynamic binding table entries with a MAC address that no longer appears in the MAC address table are cleared from the binding table. To retain these entries regardless of their MAC address table status, use the **dhcpv6-snooping binding persistency** command. For example:

```
-> dhcpv6-snooping binding persistency enable
```

When binding table retention is enabled, entries remain in the table for the term of their DHCPv6 lease and are not removed even when the MAC address for the entry is cleared from the MAC address table.

However, when the DHCP client host is connected to another port, the associated binding table is moved from the previous port to the new port.

To disable binding table retention, use the following command:

```
-> dhcpv6-snooping binding persistency disable
```

Use the **show dhcpv6-snooping** command to determine the status of binding table retention.

Adding or Modifying the Binding Table Entries

The binding table entries can be manually added, modified, and deleted. When a new binding entry is added, or an existing entry is modified, the entry's lease lifetime is changed to indefinite (that is, the entry

is converted to a static binding entry). To add or modify the binding table entries, use the **dhcpv6-snooping binding** command. For example:

```
-> dhcpv6-snooping binding vlan 1 link-local fe80::eae7:32ff:fea4:6321 global-
address 2001:db8:3001::3 mac-address 00:00:01:1d:4f:7d linkagg 1
```

Notes:

- If a VLAN is deleted, all binding entries on the VLAN including the manually added binding entry is also removed.
- While adding a new binding entry, the values for all the parameters must be specified. Else, the binding entry will not be added.

For more information about using the **dhcpv6-snooping binding** command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring IPv6 Source Filtering (ISF)

IPv6 source filtering applies to DHCPv6 Snooping ports, link aggregates, and VLANs and restricts port traffic to only packets that contain the client source MAC address, IPv6 address, and VLAN combination. The DHCPv6 Snooping binding table is used to verify the client information for the port or VLAN that is enabled for IPv6 source filtering.

IPv6 source filtering can be enabled per-VLAN or per-port (link aggregate). These two are mutually exclusive.

Notes:

- If DHCPv6 Snooping is enabled on the switch level, then ISF can be enabled on any port or VLAN.
- If DHCPv6 Snooping is enabled only on a VLAN, then ISF can only be enabled on any ports which are part of that VLAN or on the same VLAN.

Configuring IPv6 Source Filtering on an OmniSwitch 6560

To support IPv6 Source Filtering on the OmniSwitch 6560, the TCAM mode for the switch must be changed to source IPv6 filtering. By default the TCAM mode is set for destination IPv6 filtering (source IPv6 filtering is not allowed). Use the **capability profile tcam mode** with the **source-ipv6** option to change the mode to IPv6 source filtering. For example:

```
-> capability profile tcam mode source-ipv6
```

Note. Each time the TCAM mode is changed, a switch reboot is required to activate the new mode.

The following functionality is *not* supported when the source IPv6 filtering mode is active:

- Destination IPv6 source filtering.
- ISSU
- QoS anti-spoofing
- Fewer QoS policy rules supported.

To change the TCAM mode back to the destination IPv6 filtering (the default), use the **capability profile tcam mode** command with the **dest-ipv6** option. For example:

```
-> capability profile tcam mode source-ipv6
```

To verify which TCAM mode is active for the switch, use the **show capability profile** command. For example:

```
-> show capability profile
Configured  TCAM Mode :      dest-ipv6
Active  TCAM mode     :      dest-ipv6
```

Configuring Port IPv6 Source Filtering

Port source filtering is based on the interface number, source MAC address, and source IPv6 address.

By default, IPv6 source filtering is disabled for a DHCPv6 Snooping port or a link aggregate. Use the **dhcpv6-snooping ipv6-source-filter** command to enable or disable ISF for a specific port, range of ports, or a link aggregate. For example:

To enable source filtering on individual chassis and port 1/1/1, enter:

```
-> dhcpv6-snooping ipv6-source-filter port 1/1/1 admin-state enable
```

To enable source filtering on link aggregate 2, enter:

```
-> dhcpv6-snooping ipv6-source-filter linkagg 2 admin-state enable
```

To disable source filtering, enter:

```
-> dhcpv6-snooping ipv6-source-filter port 1/1/1 admin-state disable
-> dhcpv6-snooping ipv6-source-filter linkagg 2 admin-state disable
```

Configuring VLAN IPv6 Source Filtering

VLAN source filtering is based on the source VLAN ID, interface number, source MAC address, and source IPv6 address.

IPv6 source filtering can be enabled at a VLAN level and the ports associated with the VLAN when DHCPv6 Snooping is enabled at the system level or VLAN level.

By default, IPv6 source filtering is disabled for a DHCP Snooping VLAN.

Use the **dhcpv6-snooping ipv6-source-filter** command to enable or disable ISF for a VLAN. For example, to enable source filtering on VLAN 10, enter:

```
-> dhcpv6-snooping ipv6-source-filter vlan 10 admin-state enable
```

To verify the IPv6 source filtering configuration, use the **show dhcpv6-snooping ipv6-source-filter** command.

Using IPv6 DHCP Guard

DHCPv6 Guard protects the host connected to the switched network against rogue DHCPv6 servers. When this functionality is enabled, DHCPv6 server messages are discarded unless the messages are received on trusted source ports. DHCPv6 Guard functionality can also be applied to client messages to ensure that client messages are sent out only on trusted source ports.

Configuring ports as trusted sources provides a filtering mechanism to allow or drop DHCPv6 messages.

- Enabling DHCPv6 Guard and configuring trusted ports restricts DHCPv6 client and server messages to only those ports designated as trusted.
- Enabling DHCPv6 Guard without configuring any trusted ports helps to prevent unwanted DHCPv6 traffic flow through the switch. For example:
  - DHCPv6 server messages are discarded, which helps to prevent messages from reaching clients on the VLAN.
  - If DHCPv6 Guard for client messages is enabled, then DHCPv6 multicast client messages are also discarded. This helps to prevent DHCPv6 traffic from getting past the switch. If there are no client messages sent out, then there are no responses sent from the DHCPv6 server.

DHCPv6 Guard is configured on a per-VLAN basis. Make sure ports configured as trusted sources are members of the VLAN on which DHCPv6 Guard is configured.

Configuring IPv6 DHCP Guard

1 Enable DHCPv6 Guard on a VLAN or a VLAN range using the **ipv6 dhcp guard** command. For example:

```
-> ipv6 dhcp guard vlan 200 admin-state enable

-> ipv6 dhcp guard vlan 200-210 admin-state enable
```

2 To optionally enable DHCPv6 Guard for client messages, use the **ipv6 dhcp guard** command with the **client** option. For example:

```
-> ipv6 dhcp guard vlan 200 client enable
```

3 Configure switch ports or link aggregates or port range or link aggregate range as trusted DHCPv6 Guard ports using the **ipv6 dhcp guard trusted** command. For example:

```
-> ipv6 dhcp guard vlan 200 trusted port 2/1/11

-> ipv6 dhcp guard vlan 200 trusted port 2/1/11-15

-> ipv6 dhcp guard vlan 200 trusted linkagg 10
```

Note. Configure ports to which a DHCPv6 server is connected or on which a relay is configured as trusted ports to ensure that client messages will reach the server or relay agent.

Verifying the IPv6 DHCP Guard Configuration

Use the **show ipv6 dhcp guard** command to verify the IPv6 DHCP Guard configuration. For example:

```
-> show ipv6 dhcp guard
Interface                        Status   Client   Trusted Ports
---------------------------------+--------+--------+-------------------------------
VLAN 200                         Enabled  Disabled 1/1/9, 1/1/10, 1/1/11, 1/1/12+
VLAN 250                         Enabled  Disabled
VLAN 300                         Enabled  Enabled  agg 10, 1/4/20
```

To display the detailed configuration for a VLAN, use the **show ipv6 dhcp guard** command with the **vlan** option. For example:

```
-> show ipv6 dhcp guard vlan 200
DHCPv6 Guard = Enabled
Client Guard = Disabled
Trusted ports:
  1/1/9
  1/1/10
  1/1/11
  1/1/12
  1/1/20

-> show ipv6 dhcp guard vlan 300
DHCPv6 Guard = Enabled
Client Guard = Enabled
Trusted ports:
  linkagg 10
  1/4/20
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for more details on the CLI command.

Verifying the DHCPv6 Configuration

To display information about the DHCPv6, use the **show** commands listed below.

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide.*

show ipv6 dhcp relay	Displays all the interfaces on which the DHCPv6 relay is configured, the relay destinations, and the status of the DHCPv6 relay.
show dhcpv6-snooping	Displays the global DHCPv6 Snooping configuration.
show dhcpv6-snooping interfaces	Displays the DHCPv6 Snooping configuration status per-VLAN.
show dhcpv6-snooping binding	Displays the DHCPv6 Snooping binding table information.
show dhcpv6-snooping ipv6-source-filter	Displays the port, VLAN or link aggregation on which IPv6 Source Filter (ISF) is configured.
show ipv6 dhcp guard	Displays the DHCPv6 Guard configuration for an IPv6 interface.

24 Configuring an Internal DHCP Server

The Dynamic Host Configuration Protocol (DHCP) offers a framework to provide configuration information to client interfaces on an IPv4 or IPv6 IP network. DHCP is based on the Bootstrap Protocol (BOOTP) and provides additional capabilities, such as dynamic allocation of reusable network addresses and configuration options.

A DHCP server provides dynamic IP addresses on lease for client interfaces on a network. It manages a pool of IP addresses and information about client configuration parameters. The DHCP server obtains an IP address request from the client interfaces. After obtaining the requests, the DHCP server assigns an IP address, a lease period, and other IP configuration parameters, such as the subnet mask and the default gateway.

This chapter describes how to configure the internal DHCP server on the OmniSwitch.

In This Chapter

This chapter describes configuration of the DHCP server and how to modify the configuration through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details on the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- "DHCP Server Default Values" on page 24-2.
- "Quick Steps to Configure Internal DHCP Server" on page 24-2.
- "DHCP Server Overview" on page 24-4
- "Interaction With Other Features" on page 24-5
- "Configuring DHCP Server on OmniSwitch" on page 24-6
- "DHCP Server Application Example" on page 24-11
- "Configuration File Parameters and Syntax" on page 24-14
- "Policy File Parameters and Syntax" on page 24-25

DHCP Server Default Values

Parameter Description	Command	Default Value/Comments
DHCP Server operation	**dhcp-server status**	disabled

Quick Steps to Configure Internal DHCP Server

DHCP server software is installed on the OmniSwitch to centrally manage IP addresses and other TCP/IP configuration settings for clients present on a network.

Follow the steps in this section for a quick tutorial on how to configure an internal DHCP server on the OmniSwitch.

Note. For detailed information on how to configure the DHCP server on OmniSwitch, see the Configuring DHCP Server on OmniSwitch section. The *.conf and *.pcy files can be created using VitalQIP, refer to the VitalQIP documentation for additional information.

1 Navigate to **/flash/switch** directory.

```
-> cd /flash/switch
```

2 Create and customize the **dhcp.conf** and **dhcpd.pcy** files according to your requirements. Use the **vi** command to modify the existing configuration file.

Note: Both the **dhcpd.conf** and **dhcpd.pcy** files must be present for the DHCP server to be enabled.

```
-> vi dhcpd.conf
```

Declare dynamic DHCP options, global options, and server configuration parameters for client interfaces in the **dhcpd.conf** file. Add DHCP related information for a particular subnet.

For example, for the subnet 200.0.0.0, define the dynamic DHCP range, router option, domain name and other details using the following code:

```
server-identifier sample.example.com;

subnet 200.0.0.0 netmask 255.255.255.0
{
   dynamic-dhcp range 200.0.0.10 200.0.0.11
    {
            option subnet-mask 255.255.255.0;
            option routers 200.0.0.254;
            option domain-name-servers 200.0.0.99;
            option domain-name "example.com";
            option dhcp-lease-time 30000;
    }
}
```

Note. See "Configuration File Parameters and Syntax" on page 24-14 for details on what each of the optional keywords specify.

3 After entering the required information in the **dhcpd.conf** file type **`:wq`** to save the changes made to the **dhcpd.conf** file.

4 Create and customize the **dhcpd.pcy** file according to your requirements. Use the **vi** command to modify the existing configuration file.

```
-> vi dhcpd.pcy
```

For example:

```
PingAttempts=0
PingDelay=500
HonorRequestedLeaseTime=False
RegisteredClientsOnly=False
ForceClass=None
```

5 After entering the required information in the **dhcpd.pcy** file, type **`:wq`** to save the changes made to the **dhcpd.pcy** file.

Notes.

- If the **dhcpd.conf** file is corrupted, the **dhcpd.conf.lastgood** file is used as a backup file.
- If the **dhcpd.conf** file is updated successfully, then the **dhcpd.conf.lastgood** file is over written with the configurations present in the **dhcpd.conf** file.
- Properly configured **dhcpd.conf** and **dhcpd.pcy** files can be transferred to the switch remotely instead of using the vi editor.

---

6 Restart the DHCP server using the **dhcp-server restart** command. The changes made in the **dhcpd.conf** file are applied to the OmniSwitch.

```
-> dhcp-server restart
```

Note. The **dhcp-server restart** command automatically updates the **dhcpd.conf**, **dhcpd.conf.lastgood** and **dhcpd.pcy** files.

---

7 Enable the DHCP server using the **dhcp-server** command.

```
-> dhcp-server enable
```

8 Check the IP address leases by entering the following command:

```
-> show dhcp-server leases

IP address        MAC address     Lease Granted           Lease Expiry     Type
-------------+----------------+---------------------+--------------------+---------
200.255.91.53 10:fe:a2:e4:32:08 2010-01-16 11:38:47  2010-01-17 11:38:47   Dynamic
200.255.91.5  20:fe:a2:e4:32:08 2010-01-16 10:30:00  2010-01-18 10:30:00   Static
200.255.91.56 20:fe:a2:e4:33:08 2010-01-16 10:30:00   2010-01-18 10:30:00   Dynamic
200.255.91.58 20:fe:a2:e4:34:08 2010-01-16 10:30:00  2010-01-18 10:30:00   Dynamic
```

DHCP Server Overview

DHCP consists of two components:

- A protocol to supply client-specific configuration parameters from a DHCP server to a client.
- A mechanism to allocate network addresses to clients.

A DHCP server uses the Dynamic Host Configuration Protocol to provide initialization parameters to the clients in the network.

The DHCP process

DHCP is built on a client-server model, where a designated DHCP server allocates network addresses and delivers configuration parameters to dynamically configured client addresses.The process for a client to obtain its IP address through a DHCP server is as follows:

1. The client generates a DHCP request message via UDP broadcast.
2. The server listens for this request message.
3. The server responds with a DHCP reply and a valid IP address.
4. The server responds with a dynamic address in a defined range or one based on a MAC address.
5. The server leases the address for a specific time period.

Internal DHCP Server on OmniSwitch

The OmniSwitch internal DHCP server provides the abilities to:

- Enable or disable the DHCP server.
- Dynamically modify the DHCP configuration, using the **vi** editor, or through an accurately configured text file transferred to the switch.
- Restart the DHCP server.
- View the DHCP leases offered by the internal DHCP server.
- View the DHCP server statistics through the command line interface.

Note. 8K leases are supported for both DHCPv4 and DHCPv6. However, large ranges of leases are not supported. For example, it is recommended to use /112 prefix for v6-subnet in dhcpdv6.conf file to avoid memory issues.

VitalQIP Server

The VitalQIP framework provides a complete solution for IP Address Management. The OmniSwitch runs the relevant components for a remote server such as Message Service and Active Lease that interact with the QIP server. The QIP server can be used to generate the required *conf* and *pcy* files which can used on the OmniSwitch to allow it to act as a remote server.

OmniSwitch DHCP Server Management

The DHCP Server on the OmniSwitch is managed from VitalQIP by specifying the IP address of the OmniSwitch through which the VitalQIP server can manage the OmniSwitch. This can be the Loopback0 or other IP interface that is reachable by the VitalQIP server.

VitalQIP Message Service

This is the VitalQIP component that is present on the OmniSwitch acting as a remote server. This component allows the OmniSwitch to interact with other VitalQIP components. Also other services such as Active Lease Service register with the Message Server.

VitalQIP Active Lease Service

This component interacts with the VitalQIP server and provides the following capabilities:

- Viewing DHCP leases in VitalQIP
- Deleting Leases from VitalQIP.

Interaction With Other Features

This section contains important information about the internal DHCP server and its interaction with other OmniSwitch features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

Virtual Router Forwarding (VRF)

The same subnet cannot be shared across different VRFs since there is a single internal DHCP server instance.

DHCP Snooping

Internal DHCP server and DHCP snooping are mutually exclusive and cannot function together in the default VRF. DHCP snooping security is disabled when the DHCP server feature is enabled on the switch since the DHCP server is internal and secure.

IP Interfaces

The DHCP client gets a lease only if the switch has an IP interface and the DHCP server is configured for that particular subnet. If there are no IP address ranges defined for the incoming client interface, then the client is not assigned a lease.

In case of IP multinetting, the primary interface address is used to calculate the subnet of the interface. If there are no IP interfaces configured in the system, then the packet sent from the client is dropped.

VitalQIP Server

The VitalQIP framework provides a complete solution for IP Address Management. The OmniSwitch runs the relevant components for a remote server such as Message Service and Active Lease that interact with the VitalQIP server.

Configuring DHCP Server on OmniSwitch

The DHCP server implementation on OmniSwitch makes use of the policy, configuration, and server database files stored in the **/flash/switch** directory. Both the configuration and policy files must be present for the DHCP server to be enabled. The functions of the DHCP server related files are as follows:

- **DHCP Policy file:** The **dhcpd(v6).pcy** file initializes the global attributes for the DHCP server.
- **DHCP Configuration files:** The **dhcpd(v6).conf** file is used to configure specific DHCP server settings on the switch such as IP address ranges and options. The **dhcpd(v6).conf.lastgood** file is a backup for the **dhcpd(v6).conf** file.
- **DHCP Server Database file:** The **dhcpd(v6)Srv.db** file is activated only during takeover and server restart of the DHCP server. It contains lease details of the client IP addresses.

Policy file

The policy file is used to configure the DHCP related policies according to user requirements. The DHCP server policy parameters can be defined using the policy file. Ideally, most of the server parameters are kept static.

Example of a *dhcpd.pcy* File

```
PingDelay = 200
PingAttempts = 3
PingSendDelay = 1000
DefaultLease = 86400
```

Example of a *dhcpdv6.pcy* File

```
;
; QIP DHCPv6 Policy File
;
AbusiveClientMonitorPeriod=30
AbusiveClientWarningCount=30
AbusiveClientLockout=0
AddManualToGlobalDuidPool=1
AllowClientPacketsWithInvalidOptions=1
AllowUnencodedFqdn=1
CheckTransactionID=0
ClientFqdnOptionSupport=client
ClientHostNameProcessing=correct
ClientProcessingWaitTime=3000
CompressedLog=0
DefaultLease=60000
DHCPv6SocketAddr=2620:0:60:1480::3
DuidWarningsToEventLog=0
;
ForceClass=user
HonorRequestedLeaseTime=1
LogLeaseGrantAndRenew=0
```

```
MaxOutgoingDhcpMessageSize=1024
MaxPendingSeconds=120
MaxUnavailableTime=3600
MinimumRequestedLeaseTime=3600
NumberOfThreads=15

RegisteredClientsOnly=0
ReplyToUnmanagedInformationRequests=1
SendRequestedParamsOnly=1
SendUnicastOption=1
;ServerDuidDefault=0001000146e6ebb10003ba3cbb0d
ServerPreference=255
ServerStatefulMode=1
UpdateQIP=all
```

The updated **dhcpd(v6).pcy** file is effective only after the **dhcp-server restart** command is performed.

See the "Policy File Parameters and Syntax" on page 24-25 table for additional information on individual policy parameters and how to apply the policies for internal DHCP server on the OmniSwitch.

DHCP Configuration Files

The configuration files store the network information for the DHCP clients. There are two main DHCP configuration files that can be used to configure the DHCP server on OmniSwitch. They are:

- **dhcpd(v6).conf**
- **dhcpd(v6).conf.lastgood**

The following sections provide detailed information on the **dhcpd(v6).conf** and **dhcpd(v6).conf.lastgood** files.

dhcpd(v6).conf File

The **dhcpd(v6).conf** file is used to declare DHCP options and global options for the DHCP clients. The **dhcpd(v6).conf** can be used to define the following:

- IP subnets
- Dynamic scopes and static bindings
- Subnet masks, DNS and default routers, and lease times
- User class or vendor class configurations

There are three types of statements in the configuration file:

- **Parameters:** Declare how, when, or what to provide to a client.
- **Declarations**: Describe the topology of the network and provide addresses for the clients. Parameters can be listed under declarations that override the global parameters.
- **Comments**: Provide a description for the parameters and declarations. Lines beginning with a hash mark (#) are considered comments and they are optional.

Example *dhcpd.conf* File

```
#Global parameters that specify addresses and lease time.
option domain-name-servers 200.0.0.99;
option domain-name "example.com";
option dhcp-lease-time 20000;

#IP subnet
subnet 200.0.0.0 netmask 255.255.255.0
{
  #Dynamic scope and parameters that apply to this scope overriding global params.

dynamic-dhcp range 220.0.0.100 220.0.0.130
  {
    option routers 220.0.0.254;
    option subnet-mask 255.255.255.0;
    option domain-name "scope_example.com";
    option domain-name-servers 192.168.1.1;
    option dhcp-lease-time 30000;
  }

 #Static binding based on MAC address
  manual-dhcp 00-01-02-03-04-05 220.0.0.140
  {
    option subnet-mask 255.255.255.0;
  }
}
```

Note. A subnet declaration must be included for every subnet in the network related to the DHCP server.

Details about valid parameters and declarations are listed in the table found in "Configuration File Parameters and Syntax" on page 24-14.

Note. The IPv6 address range prefix in the **dhcpdv6.conf** file is recommended to be not less than /96. With a less specific mask there may be an increase in memory allocation/utilization.

Example *dhcpdv6.conf* File

```
v6-server-identifier company.example.com;

     duid-pool { <- DUID pool for which we allocate IP across sunbets
         00-03-02-be-1a-0f-cd-14-67-98-05-56-98-98-67-cd-69-01
         00-02-00-00-00-09-*
         00-02-00-00-00-08-*
         00-02-00-00-00-07-*
         00-01-*
     }
     x-duid-pool { <- Excluded DUID pool for which we do not allocate IP
         00-02-00-00-00-09-0c-c0-84-d3-03-00-09-12
     }
```

```
     v6-subnet 2620:0000:0060:1480:0000:0000/97 { <- IPV6 subnet
         x-duid-pool { <- Excluded DUID pool for this subnet
             00-02-00-00-00-09-0c-c0-84-d3-03-00-09-13
             00-02-00-00-00-09-0c-c0-84-d3-03-00-09-19
         }
         policy send-unicast-option-enabled false; <- policy options applicable
         policy subnet-unavailable-threshold 90;
         policy subnet-unavailable-descent-threshold 85;
         policy minimum-requested-lifetime 800;
         option renewal-time 700000; <- Options applicable
         option rebinding-time 1000000;
         option preferred-lifetime 2000000;
         option valid-lifetime 3000000;
         option dns-recursive-name-server 2001:468:603:c0e0::12001:468:603:c0e0::2;

         v6-manual-dhcp duid 00-02-00-00-00-09-0c-c0-84-d3-03-00-0a-11
2620:0000:0060:1480::1f01 { <- Manual DUID mapping
            option posix-timezone "MST7MDT6,116/02:00:00,298/02:00:00";
         }

         v6-dynamic-dhcp range 2620:0000:0060:1480::2000 2620:0000:0060:1480::2500
{ <- Dynamic range of IPs
            policy minimum-requested-lifetime 650;
    policy rapid-commit-enabled true;
            policy  excluded-user-classes "bronze" "gold" "silver";
            policy excluded-vendor-classes enterprise 311 "MSFT 5.0" enterprise 546
"SIP Phone";
            option dns-recursive-name-server 2620:0000:0060:1480::3
2620:0000:0060:1480::4;
            option domain-search-list malvern2.lucent.com murrayhill2.lucent.com;
            option posix-timezone "CST6CDT5,116/02:00:00,298/02:00:00";
            option sntp-servers 2620:0000:0060:1480::5 2620:0000:0060:1480::6;
         }
     }

     v6-subnet 2620:0000:0060:1481/64 {
         policy minimum-requested-lifetime 800;
         v6-manual-dhcp duid 00-02-00-00-00-09-0c-cd-84-d3-03-00-0a-14
2620:0000:0060:1481::1f01 {
            option posix-timezone "MST7MDT6,116/02:00:00,298/02:00:00";
         }
         v6-manual-dhcp duid 00-02-00-00-00-09-0c-cd-84-d3-03-00-0a-13 iaid 1001
2620:0000:0060:1481::1f02 {
            option posix-timezone "MST7MDT6,116/02:00:00,298/02:00:00";
         }
         v6-manual-dhcp duid 00-02-00-00-00-09-0c-cd-84-d3-03-00-0a-13 iaid 1002
2620:0000:0060:1481::1f03 {
            option posix-timezone "MST7MDT6,116/02:00:00,298/02:00:00";
```

dhcpd(v6).conf.lastgood File

The **dhcpd(v6).conf.lastgood** file is used as a backup file when the **dhcpd(v6).conf** file is corrupted. If the **dhcpd(v6).conf** file is affected, then the DHCP server generates an error. In such an instance, the DHCP server configuration is updated according to the **dhcpd(v6).conf.lastgood** file. The **dhcpd(v6).conf.lastgood** file is now used to configure the internal DHCP server, provide IP addresses on lease, and maintain DHCP related information.

The **dhcpd(v6).conf.lastgood** file is overwritten with the configurations in the **dhcpd(v6).conf** file when the DHCP configurations are setup or updated and the internal DHCP server is restarted successfully. At this point, the **dhcpd(v6).conf** and **dhcpd(v6).conf.lastgood** files are identical.

If any modifications are made in the **dhcpd(v6).conf** file, the DHCP server must be restarted so that the configuration is updated on the OmniSwitch. The **dhcp-server restart** command automatically updates the **dhcpd(v6).conf** and **dhcpd(v6).conf.lastgood** files.

DHCP Server Database file

The **dhcp(v6)Srv.db** or the DHCP server database or lease file is initialized when the DHCP server function takes over or is restarted. The DHCP server database file contains the mappings between a client IP address and MAC address, referred to as a binding.

There are two types of bindings:

Static bindings - Map a single MAC address to a single IP address.

Dynamic bindings - Dynamically map a MAC address to an IP address from a pool of IP addresses. Details of both the dynamic and static bindings, are stored in the **dhcp(v6)Srv.db** file.

The **dhcp(v6)Srv.db** file is read when the switch reloads or the DHCP service restarts. The server database file is read-only and must not be opened or edited by the user. This file provides an account of all the subnets configured and helps in detecting all the unmanaged leases. The lease file is synchronized with the DHCP server periodically based on a timer for smooth operation during takeover and restart. The default value of this timer is 1 minute. The timer ping mechanism is used to prevent duplicate IP address allocations to clients in the same subnet. The lease file synchronization is applicable for both chassis and stack based OmniSwitch products.

Flushing the DHCP Server Database file

Follow the steps below to completely flush the lease file as needed. The process is the same for both IPv4 and IPv6 DHCP servers, use the appropriate CLI commands.

1 Disable the DHCP server.

```
-> dhcp-server disable
-> dhcpv6-server disable
```

2 Delete the lease file.

```
-> rm /flash/switch/dhcpsrv.db
-> rm /flash/switch/dhcpv6srv.db
```

3 Restart the DHCP server.

```
-> dhcp-server restart
-> dhcpv6-server restart
```

4 -Enable the DHCP server.

```
-> dhcp-server enable
-> dhcpv6-server enable
```

DHCP Server Application Example

In this application example the clients or hosts obtain their IP addresses from the internal DHCP server configured on the OmniSwitch. DHCP clients initially have no IP address and are provided IP addresses by the DHCP server.

The external router supports the DHCP relay functionality so that it can forward DHCP frames sent to and from the DHCP clients and server on the OmniSwitch.

In the diagram on page 24-12, the OmniSwitch is acting as a DHCP server and the external router is acting as the DHCP relay agent. The DHCP requests from the clients (eg: 200.0.0.X) are relayed from the external router to the OmniSwitch acting as a DHCP server. The internal DHCP server on OmniSwitch processes the requests and leases IP addresses based on the DHCP server configuration.

1 The DHCP clients are present in the 200.0.0.X network connected to the external router and also in the 220.0.0.X network directly attached to the OmniSwitch.

2 The default **dhcpd.pcy** file can be used to configure the DHCP server global parameters.

3 The **dhcpd.conf** file defines the 200.0.0.X network and 220.0.0.X network.

4 The subnet mask and DNS server address are global declarations since they are the same for each subnet.

5 The default router address and lease times are declared as a part of the scope since they are different for each subnet.

6 The resulting sample code for the **dhcpd.conf** file is as follows:

```
#Global parameters
option subnet-mask 255.255.255.0;
option domain-name-servers 200.0.0.99;
subnet 200.0.0.0 netmask 255.255.255.0
{
  dynamic-dhcp range 200.0.0.11 200.0.0.20
  {
    option routers 200.0.0.254;
    option dhcp-lease-time 20000;
  }
}

subnet 220.0.0.0 netmask 255.255.255.0
{
  dynamic-dhcp range 220.0.0.100 220.0.0.105
  {
    option routers 220.0.0.254;
    option dhcp-lease-time 30000;
  }

}
```

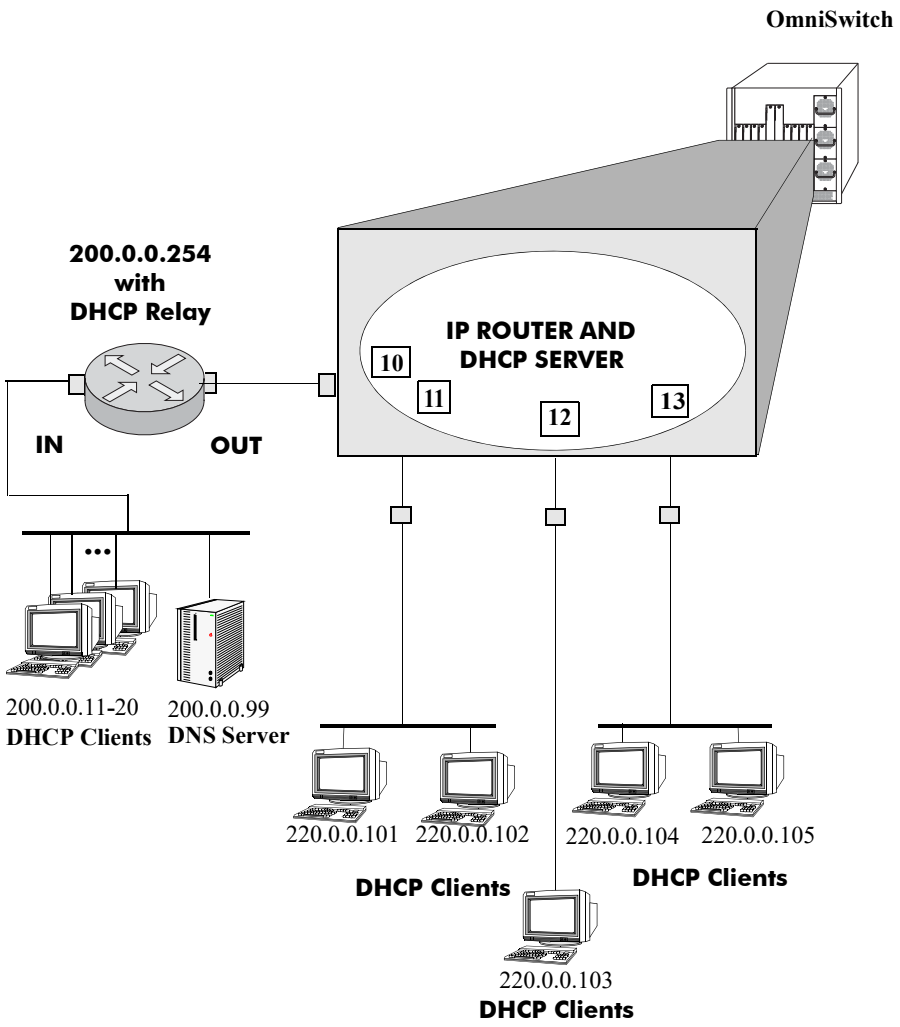


Figure 24-1 : Internal DHCP Server Application Example

Verifying the DHCP Server Configuration

To display information about the DHCP Server configuration and statistics use the **show** commands listed below:

show dhcp-server leases	Displays the leases offered by the DHCP server.
show dhcp-server statistics	Displays the statistics of the DHCP server.

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration File Parameters and Syntax

The following table provides detailed information about the configuration file options and syntax specifications.

Option Code	dhcpd.conf Key-word	dhcpd.conf example	Data type	Default Value	Description
1	subnet-mask	`option subnet-mask 255.255.0.0;`	N/A	Same as in Subnet Profile	Specifies the client's subnet mask. If both the subnet mask and the router option are specified in a DHCP reply, the subnet mask option must be specified before the router option.
2	time-offset	`option time-offset 1000;`	numeric_ signed	N/A	Specifies the offset of the client's subnet (in seconds) from Coordinated Universal Time (also referred to as UTC). A positive offset indicates a location east of the zero meridian and a negative offset indicates allocation west of the zero meridian. For example, to enter a time offset for a client subnet located in the Eastern Standard Timezone (5 hours west of the UTC zero meridian), enter -18000.
3	routers	`option routers 100.0.0.1;`	N/A	Same as in Subnet Profile	Lists the IP addresses for the routers for each client subnet defined. Routers should be listed in order of preference.
4	time-server	`option time-server 10.10.0.10;`	N/A	Same as in Subnet Profile	Specifies IP address of the RFC 868 time server available to the client.
5	name-servers	`option name-servers 10.10.0.100;`	ip_address_ list	N/A	Specifies IP address of the IEN-116 name server available to the client.
6	domain-name-servers	`option domain-name-servers 10.10.0.30;`	N/A	Same as in Subnet Profile	Lists the DNS (STD 13, RFC 1035) name server IP address(es) available to the client. Servers should be listed in order of preference.
7	log-servers	`option log-servers 10.10.0.100;`	ip_address_ list	N/A	Specifies the IP address of the MIT-LCS UDP log server available to the client.
8	cookie-servers	`option cookie-servers 10.10.0.100;`	ip_address_ list	N/A	Specifies the IP address of the RFC 865 cookie server available to the client.
9	lpr-servers	`option lpr-servers 10.10.0.100;`	ip_address_ list	N/A	Specifies IP address of the line printer server available to the client.
10	impress-servers	`option impress-servers 10.10.0.100;`	ip_address_ list	N/A	Specifies IP address of the Imagen Impress server available to the client.

Option Code	dhcpd.conf Key-word	dhcpd.conf example	Data type	Default Value	Description
11	resource-location-servers	`option resource-location-servers 10.10.0.100;`	ip_address_list	N/A	Specifies the IP address of the Resource Location server available to the client.
12	host-name	`option host-name "bgp00001 4bgs";`	N/A	Same as in Object Profile	Specifies the name of the client. If the host name is defined in an option template, it overrides any definition in the Object Profile.
13	boot-size	`option boot-size 30;`	numeric	N/A	Specifies the length of the default boot image of the client. The maximum file length is 65,535 bytes.
14	merit-dump	`option merit-dump "m_dump";`	text	N/A	Specifies the pathname of the file where the core image is to be dumped in the occurrence of a crash. The path is formatted as a character string consisting of characters from the Network Virtual Terminal (NVT) ASCII character set.
15	domain-name	`option domain-name "abc.example.com";`	N/A	Same as in Subnet Profile	Specifies the domain name to resolve hostnames via the Domain Name Service (DNS).
16	swap-server	`option swap-server 10.10.0.100;`	ip address	N/A	Specifies the IP address of the client's swap server.
17	root-path	`option root-path "/root";`	text	N/A	Specifies the pathname that contains the client's root directory or partition. The path is formatted as an NVT ASCII character string.
18	extensions-path	`option extensions-path "/ext";`	text	N/A	Specifies a text string to denote a file, retrievable via Trivial File Transfer Protocol (TFTP). The file contains information that can be interpreted in the same way as the 64-octet vendor-extension field within the BOOTP response. The length of the file is unconstrained. All references to instances of the BOOTP Extensions Path field within the file are ignored.
19	ip-forwarding	`option ip-forwarding false;`	boolean	False	Select **True** to configure the IP layer to enable packet forwarding. Select **False** to disable packet forwarding.
20	non-local-source-routing	`option non-local-source-routing false;`	boolean	False	Select **True** to configure the IP layer to forward datagrams with non-local source routes. Select **False** to disable forwarding of the datagrams.

Option Code	dhcpd.conf Key-word	dhcpd.conf example	Data type	Default Value	Description
21	policy-filter	`option policy-filter 10.10.0.100 255.255.0.0;`	ip_ address_ mask_list	N/A	Specifies policy filters for nonlocal source routing. The filters consist of the IP address list and masks. This data specifies destination and mask pairs with which to filter incoming source routes. The client should discard any source-routed datagram whose next hop address does not match one of the filters.
22	max-dgram-reassembly	`option max-dgram-reassembly 576;`	numeric	N/A	Specifies the maximum reassembly size of the datagram. Enter a value between 576 and 65,535.
23	default-ip-ttl	`option default-ip-ttl 1;`	numeric	N/A	Specifies the default time-to-live (in seconds) to use on outgoing datagrams as an octet between 1 and 255.
24	path-mtu-aging-timeout	`option path-mtu-aging-timeout 10;`	numeric	N/A	Specifies the maximum time to be allotted for Path Maximum Transmit Unit (MTU) values to be discovered. The timeout is in seconds, from 0 to 2,147,483,647.
25	path-mtu-plateau-table	`option path-mtu-plateau-table 68;`	numeric_list	N/A	Identifies a table of MTU sizes to use when performing Path MTU discovery as defined in RFC 1191. The table is formatted as a list. Minimum value is 68. Maximum value is 65,535.
26	interface-mtu	`option interface-mtu 68;`	numeric	N/A	Specifies the Maximum Transmit Unit (MTU) to be used on the related interface. MTU is the frame size in a TCP/IP network. Valid range from 68 to 65,535.
27	all-subnets-local	`option all-subnets-local false;`	boolean	False	**True** indicates that all subnets share the same MTU as of the subnet to which the client user is directly connected **False** indicates that some of the subnets connected may have smaller MTUs.
28	broadcast-address	`option broadcast-address 10.10.255.255`	N/A	Same as in Subnet Profile	Specifies the broadcast address used on the client's subnet.
29	perform-mask-discovery	`option perform-mask-discovery false;`	boolean	False	**True** indicates that the client should perform subnet mask discovery. **False** indicates that no mask discovery must be performed.

Option Code	dhcpd.conf Key-word	dhcpd.conf example	Data type	Default Value	Description
30	mask-supplier	`option mask-supplier false;`	boolean	False	**True** indicates that response to the subnet mask request should use Internet Control Message Protocol (ICMP). **False** indicates the subnet mask should not respond using ICMP.
31	router-discovery	`option router-discovery false;`	boolean	False	**True** allows router discovery to be performed as defined in RFC 1256. **False** indicates that router discovery need not be performed.
32	router-solicitation-address	`option router-solicitation-address 10.10.0.100;`	ip_address	N/A	Specifies the IP address where router solicitation requests should be transmitted.
33	static-routes	`option static-routes 10.10.0.100 10.10.0.200;`	ip_address_pair_list	N/A	Specifies the list of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority. The routes consist of a list of IP address pairs. The first address is the router for the destination. The default route (0.0.0.0) is an illegal destination for a static route.
34	"trailer-encapsulation	`option trailer-encapsulation false;`	boolean	False	Select **True** to identify whether the client should negotiate the use of trailers (RFC 893) when using the Address Resolution Protocol (ARP) . Select **False** to prevent the use of trailers.
35	arp-cache-timeout	`option arp-cache-timeout 10;`	numeric	N/A	Specifies the time-out in seconds for ARP cache entries, from 0 to 2,147,483,647.
36	ieee802-3-encapsulation	`option ieee 802-3-encapsulation false;`	boolean	False	Use this option to identify the use of Ethernet Version 2 (RFC 894) or IEEE 802.3 (RFC 1042) encapsulation for Ethernet interface. Select **True** to use RFC 1042 encapsulation. Select **False** to use RFC 894 encapsulation.
37	default-tcp-ttl	`option default-tcp-ttl 1;`	numeric	N/A	Defines the default time-to-live (in seconds) to use when sending TCP segments. Enter a value from 1 to 255.

Option Code	dhcpd.conf Key-word	dhcpd.conf example	Data type	Default Value	Description
38	tcp-keepalive-interval	`option tcp-keepalive-interval 10;`	numeric	N/A	Specifies the amount of time, in seconds, to wait before sending a keep alive message on a TCP connection. A value of 0 indicates keep alive messages on connections should not be generated unless specifically requested to do so by an application. Valid range from 0 to 2,147,483,647
39	tcp-keepalive-garbage	`option tcp-keepalive-garbage false;`	boolean	False	Specifies if the TCP keep alive messages should be sent with a garbage octet for compatibility with older implementations. Select **True** to enable a garbage octet to be sent. Select **False** to prevent a garbage octet being sent.
40	nis-domain	`option nis-domain "abc.example.com";`	text	Same as in Subnet Profile	Network Information Service (NIS) support is provided on SunOS 4.1x, Solaris 2.x and HP_UX10 only. Specify the NIS domain name. The domain is formatted as a character string from the NVT ASCII character set.
41	nis-servers	`option nis-servers 10.10.0.30;`	ip_address_list	Same as in Subnet Profile	Lists the IP addresses (in order of preference) identifying the NIS (Network Information Service) servers available to the client
42	ntp-servers	`option ntp-servers 10.10.0.50`	ip_address_list	Same as in Subnet Profile	Lists the IP addresses (in order of preference) indicating NTP (RFC 868) servers available to the client.
43	vendor-specific	`option vendor-specific vspInfo;`	hexadecimal_text	N/A	Used by clients and servers to exchange vendor-specific information. The value for this option is defined in the hexadecimal format. The definition of this information is vendor specific. The vendor is indicated in the vendor class identifier option. Servers not equipped to interpret the vendor specific information sent by a client must ignore the related data. Clients that do not receive desired vendor-specific information should attempt to operate without the related data. The clients must announce that they are working in a degraded mode.

Option Code	dhcpd.conf Key-word	dhcpd.conf example	Data type	Default Value	Description
44	netbios-name-servers	`option netbios-name-servers 10.10.0.100;`	ip_address_list	N/A	Specifies a list of RFC 1001/1002 NBNS name servers listed in order of preference. This is a NetBIOS name server (NBNS) or WINS server option.
45	netbios-dd-servers	`option netbios-dd-servers 10.10.0.100;`	ip_address_list	N/A	Specifies a list of RFC 1001/1002 NBDD servers listed in order of preference. This is a NetBIOS datagram distribution server (NBDD) option.
46	netbios-node-type	`option netbios-node-type 1;`		N/A	Allows NetBIOS over TCP/IP clients, which are configurable as described in RFC 1001/1002. The value is specified as a single octet, which identifies the client type, as follows:-ValueNode type 0x1B-node 0x2P-node 0x4M-node 0x8H-node
47	netbios-scope	`option netbios-scope "xyz";`	text	N/A	This NetBIOS scope option specifies the NetBIOS over TCP/IP scope parameter for the client, as specified in RFC 1001/1002.
48	font-servers	`option font-servers 10.10.0.100;`	ip_address_list	N/A	Specifies a list of X Window System Font servers available to the client. Servers should be listed in order of preference.
49	x-display-manager	`option x-display-manager 10.10.0.100;`	ip_address_list	N/A	Specifies a IP address list of systems that are running the X Window System Display Manager and are available to the client.
51	dhcp-lease-time	`option dhcp-lease-time 4294967295;`	time_interval	Unlimited	Used in a client request (DHCPDISCOVER or DHCPREQUEST) to allow the client to request a lease time for the IP address. In a server reply (DHCPOFFER), a DHCP server uses this option to specify the lease time offered. Selecting the Limited option allows you to set a lease time of up to 999 days, 999 hours, and 999 minutes.

Option Code	dhcpd.conf Key-word	dhcpd.conf example	Data type	Default Value	Description
52	dhcp-option-overload	`option dhcp-option-overload 1;`	1, 2 or 3	N/A	Used to indicate that the DHCP server name or file fields are being overloaded by using them to carry DHCP options. A DHCP server inserts this option if the returned parameters exceed the usual space allotted for options. If this option is present, the client interprets the specified additional fields after it concludes the interpretation of the standard option fields. Legal values for this option are as follows: 1 - The “file” field is used to hold options 2 - The “sname” field is used to hold options 3 - Both fields are used to hold options
58	dhcp-renewal-time	`option dhcp-renewal-time 10;`	numeric	N/A	Specifies the time interval from address assignment until the client transitions to the renewing state. You can enter any value from 0 to 999,999,999 seconds.
59	dhcp-rebinding-time	`option dhcp-rebinding-time 10;`	numeric	N/A	Specifies the time interval from address assignment until the client transitions to the rebinding state. You can enter any value from 0 to 999,999,999 seconds.
61	dhcp-client-identifier	`option dhcp-client-identifier xyz;`	text	N/A	Used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. It is unique for all clients in an administrative domain.The client identifier consists of type-value pairs. Ex: A hardware type and hardware address. In this case, the type field should be one of the Address Resolution Protocol (ARP) hardware types defined in RFC 1700. A hardware type - 0 indicates a domain name. Vendors and system administrators are responsible for choosing the unique client-identifiers.

Option Code	dhcpd.conf Key-word	dhcpd.conf example	Data type	Default Value	Description
62	novell-netware-domain-name	`option novell-netware-domain-name "xyz";`	text	N/A	Used to convey the NetWare/IP domain name used by the NetWare/IP product. The NetWare/IP Domain in the option is a Network Virtual Terminal (NVT) ASCII text string. You can enter up to 255 characters.
63	novell-netware-info	`option novell-netware-info [0100];`	sub-option	N/A	This NetWare/IP option code is used to convey all the NetWare/IP related information except for the NetWare/IP domain name. If NWIP_EXIST_IN_OPTIONS _AREA sub-option is set, one or more of the other suboptions may be present.
64	dhcp-nis+-domain	`option dhcp-nis+-domain "xyz";`	text	Same as in Subnet profile	Specifies the NIS domain name. The domain is formatted as a character string from the NVT ASCII character set Network Information Service (NIS) support is provided on SunOS 4.1x, Solaris 2.x and HP_UX10 only.
65	dhcp-nis+-servers	`option dhcp-nis+-servers 10.10.0.100;`	ip_address_list	Same as in Subnet profile	Lists the IP addresses identifying the NIS servers available to the client in order of preference
66	dhcp-tftp-server	`option dhcp-tftp-server "xyz";`	text	N/A	Used to identify a Trivial File Transfer Protocol (TFTP) server when the server name field in the DHCP header has been used for DHCP options.
67	dhcp-bootfile-name	`option dhcp-bootfile-name "xyz";`	text	N/A	This option is used to identify a bootfile when the file field in the DHCP header has been used for DHCP options.
68	dhcp-mobile-ip-home-agent	`option dhcp-mobile-ip-home-agent 10.10.0.100;`	ip_address_list	N/A	This option specifies an IP address list indicating mobile IP home agents available to the client. Agents should be listed in order of preference.
69	dhcp-smtp-server	`option dhcp-smtp-server 10.10.0.100;`	ip_address_list	N/A	Specifies a list of SMTP servers available to the client. Servers should be listed in order of preference.
70	dhcp-pop3-server	`option dhcp-pop3-server 10.10.0.100;`	ip_address_list	N/A	Specifies a list of POP3 servers available to the client. Servers should be listed in order of preference.
71	dhcp-nntp-server	`option dhcp-nntp-server 10.10.0.100;`	ip_address_list	N/A	This Network News Transport Protocol (NNTP) server option specifies a list of NNTP servers available to the client. Servers should be listed in order of preference.

Option Code	dhcpd.conf Key-word	dhcpd.conf example	Data type	Default Value	Description
72	dhcp-www-server	`option dhcp-www-server 10.10.0.100;`	ip_address_list	N/A	Specifies a list of WWW servers available to the client. Servers should be listed in order of preference.
73	dhcp-finger-server	`option dhcp-finger-server 10.10.0.100;`	ip_address_list	N/A	Specifies a list of Finger servers available to the client. Servers should be listed in order of preference.
74	dhcp-irc-server	`option dhcp-irc-server 10.10.0.100;`	ip_address_list	N/A	Specifies a list of IRC servers available to the client. Servers should be listed in order of preference.
75	dhcp-streettalk-server	`option dhcp-streetalk-server 10.10.0.100;`	ip_address_list	N/A	Specifies a list of StreetTalk servers available to the client. Servers should be listed in order of preference.
76	dhcp-stda-server	`option dhcp-stda-server 10.10.0.100;`	ip_address_list	N/A	Specifies a list of STDA (StreetTalk Directory Assistance) servers available to the client. Servers should be listed in order of preference.
78	slp-directory-agent	`option slp-directory-agent [000a0a0064];`	sub-option		Specifies the location of one or more SLP Directory Agents. The SLP Directory Agent option contains the following suboptions:
	Mandatory		boolean	False	This sub-option may be set to either True or False. If it is set to True, the SLP UserAgent or Service Agent so configured must not employ either active or passive multicast discovery of Directory Agents.
	Directory Agent Address		ip_address_list	N/A	This sub-option allows a IP address list to be specified. The list must be in order of preference, if an order of preference is desired.
79	slp-service-scope	`option slp-service-scope [0078797a];`	sub-option		Specifies the scopes that a SLP Agent is configured to use. It contains the following suboptions: If set to **False**, static configuration takes precedence over the DHCP provided scope list. If set to **True**, the entries in the Scope List must be used by the SLP Agent.
	Scope Listtext			N/A	This sub-option is a comma-delimited list of scopes. The list is case insensitive.

Option Code	dhcpd.conf Key-word	dhcpd.conf example	Data type	Default Value	Description
	Mandatory		boolean	FALSE	This sub-option determines whether SLP Agents override their static configuration for scopes in the Scope List. This allows DHCP administrators to implement a policy of assigning a set of scopes to Agents for service provision.
85	novell-nds-servers	`option novell-nds-servers 10.10.0.100;`	ip_address_list	N/A	Specifies one or more NDS servers for the client to contact for access to the NDS database. Servers should be listed in order of preference.
86	novell-nds-tree-name	`option novell-nds-tree-name "xyz";`	text	N/A	Specifies the name of the NDS tree which the client can contact. Maximum 255 characters.
87	novell-nds-context	`option novell-nds-context "xyz";`	text	N/A	Specifies the initial NDS context the client should use. Maximum 255 characters.
88	broadcast-multicast-service-domain	`option broadcast-multicast-service-domain [0378797a00];`	name_list	N/A	Lists server names that host the Broadcast and Multicast services that are specified as domain names.
89	broadcast-multicast-service-address	`option broadcast-multicast-service-address 10.10.0.100;`	ip_address_list	N/A	Lists server names that host the Broadcast and Multicast services that are specified as IPV4 addresses.
98	user-authentication-protocol	`option user-authentication-protocol [78797a];`	text_list	N/A	Specifies a list of Uniform Resource Locators (URLs), each pointing to a user authentication service that is capable of processing authentication requests encapsulated in the UAP. UAP servers can accept either HTTP 1.1 or SSLv3 connections.If the list includes a URL that does not contain a port component, the normal default port is assumed (port 80 for http andport 443 for https). If the list includes a URL that does not contain a path component, the path /uap is assumed.
100	timezone-posix	`option timezone-posix "xyz";`	text	255	Specifies a DHCP client's timezone specified as a POSIX 1003.1 timezone string.
101	timezone-database	`option timezone-database "xyz";`	text	255	Specifies a DHCP client's timezone specified as a TZ database string.

Option Code	dhcpd.conf Key-word	dhcpd.conf example	Data type	Default Value	Description
116	ipv4-auto-configuration	`option ipv4-auto-configuration false;`	boolean	False	This option is used to check whether, and be notified if, autoconfiguration should be disabled on the local subnet. When a server responds with the value “AutoConfigure” (True), the client may generate a linklocal IP address if appropriate. However, if the server responds with “DoNotAutoConfigure” (False), the client must not generate a link-local IP address, possibly leaving it with no IP address.
119	domain-search	`option domain-search [0378797a00];`	text_list	N/A	Passes the domains in the search list from the DHCP Server to the DHCP Client to use when resolving hostnames using DNS.
120	sip-server	`option sip-server [010a0a0064];`	ip_address_list	N/A	Lists the SIP servers specified as IPV4
121	classless-static-route	`option classless-static-route 16.10.10 10.10.0.200;`	ip_mask_ip_list	N/A	Specifies one or more static routes, each of which consists of a destination descriptor (the subnet address and subnet mask) and the IP address of the router that should be used to reach that destination.

122	cablelabs-client-config	`option cablelabs-client-config [01040a0a006402040a0a0064040c0000000a0000000a0000000a050c0000000a0000000a06050358595a00070100080100];`
		
		`option 177 [010378797a020378797a030378797a040378797a050378797a060378797a07010008010009 0378797a];`
			sub_option		The following table describes the CableLabs Client Configuration 122 sub-options, specified in RFC 3495:

Policy File Parameters and Syntax

Num	Policy	Usage	Default Value	Description
1	ActiveLease Expiration	`ActiveLease Expiration = On`	Off	Determines how the expired leases are handled. The following values are available: **Off** - prevents expired leases from being automatically deleted after lease period is over. **Full_delete** - causes the lease from DHCP database to be deleted, and the Message Service to be notified of expired leases.
2	Check TransactionID	`Check Transaction ID=True`	False	Configures the service to ignore multiple discover, request, and BootP messages that have the same XID.
3	DefaultLease	`Default Lease=86400`	7776000 (90 days)	Specifies the default lease period provided for the clients in seconds.
4	DropAll DhcpInform Packets	`DropAll Dhcp Inform Packets = True`	False	Allows administrators to configure the DHCP server to ignore inform packets. If this policy is set to **True**, the DHCP server prevents the processing of DHCPINFORM packets. However, the incoming packets are parsed.
4	DropZero MacAddress Packets	`DropZero MacAddress Packets = False`	True	If this policy is set to **True**, the DHCP server checks all incoming packets for a zero MAC address and drops the packet if it is found. **Note:** DHCPINFORM messages are processed even if this process is set to true.
5	ForceClass	`ForceClass =VendorNone`	True	Determines if the service verifies the lease request from the client before issuing a lease. The values associated with this policy are as follows: • **None** - Allows the server to issue leases from any IP address range to an incoming client request. • **Both** - Forces the service to match for both user and vendor class with the values defined for a particular IP address range. • **Vendor** - The service must match only on the vendor class. • **User** - The service must match only on user class.
6	Honor Requested LeaseTime	`Honor Requested Lease Time = False`	True	If this policy is set to **True**, the DHCP server provides the requested lease time to the client. If this policy is set to **False,** the server offers the configured lease time.

Num	Policy	Usage	Default Value	Description
7	Lease Expiration SleepTime	`Lease Expiration SleepTime = 120000`	60000 msecs	Specifies the time interval in milli seconds after which the lease expiration processing occurs. **Note:** This value must not be less than 1 minute.
8	MaxPending Seconds	`MaxPending Seconds = 20`	10	Specifies the number of seconds that an offered lease remains in a pending state. When a client sends a DHCPDISCOVER request, the DHCP server responds with a DHCPOFFER and offers an IP address. The address is marked as pending for the specified period of time.
9	Max Unavailable Time	`Max Unavailable Time = 14000`	86400	Determines the period of time that an IP address is not available after a DHCPDECLINE or ping packet is sent as response. After this time period, the server considers this address as available.
10	Nak Unknown Clients	`NakUnknown Clients = False`	True	Prevents the DHCP server from providing DHCP addresses to clients which are not in the defined subnets of the DHCP server. This policy must be set to **False** in environments where multiple DHCP services are active in the same subnet or subnets.
11	NackDhcp RequestsFor Duplicates	`NackDhcp RequestsFor Duplicates = False`	True	If this value is set to **True**, the DHCP server sends a NAK if a RENEW/REBIND request or SELECTING request is received for an IP address already owned by another hardware interface. If this value is set to **False**, the invalid request is dropped.
12	PingAttempts	`Ping Attempts = 3`	1	Specifies the number of attempts to ping through which DHCP server determines if the IP address is already in use.
13	PingDelay	`PingDelay = 200`	N/A	Specifies the delay in milliseconds between two consecutive pings to check the IP address usage in the network.
14	PingSendDelay	`PingSend Delay = 1000`	500	Specifies the number of milliseconds between subsequent pings. This is applicable only if the ping attempts are greater than 1. If the value of PingAttempts is greater than 1, then the PingSendDelay overrides the PingDelay policy.
15	PingRetention	`Ping Retention = 200`	0	Specifies the number of seconds for which a ping is valid. If a ping is attempted and no response is returned, then the address is considered to be available. During the ping retention period, other ping requests are ignored.

Num	Policy	Usage	Default Value	Description
18	PingBefore ManualDhcp	`PingBefore ManualDhcp = False`	True	If this value is set to **True**, the DHCP server performs a ping before assigning a static DHCP address. If an ICMP_REPLY is received from the ping, then the DHCP offer is not sent to the client and the address is marked as unavailable.
19	PingBefore ManualBootp	`PingBefore ManualBootp = True`	False	If this value is set to **True**, the DHCP server performs a ping before assigning a static BootP address. If an ICMP_REPLY is received, the BootP reply is not sent to the client, and the BootP address is marked as unavailable.
20	Registered ClientsOnly	`Registered Clients Only = True`	False	This poilcy is used when the MAC pool addresses are defined at either the global or the subnet level. If this value is set to **True**, the DHCP information is provided to the clients that have a known MAC address (configured in a MAC pool). If MAC pool addresses are not defined at either the global or the subnet level, the none of the devices are provided a DHCP lease. If this value is set to **False**, the DHCP information is provided to all clients.
21	SendRequested ParamsOnly	`Send Requested Params Only = True`	False	If this value is set to **True**, the DHCP server sends only the options requested by the client. For example, if the client sends a DHCP parameter request list - option (55) in the Discover packet, then the server sends only the options that are both configured and requested by the client. The subnet-mask (1) and lease-time (51) options are always sent to the client, in addition to the IP address. If this value is set to **False**, the service sends all the configured options to the client.
22	SupportRelay AgentDevice Class	`SupportRelay AgentDevice Class = True`	False	If this policy is set to **True**, the server supports the assignment of DHCP options by the DOCSIS device class.
23	ZeroCiAddr	`ZeroCiAddr = True`	False	This policy affects the contents of the “ciaddr” field in outgoing packets. If this policy is set to **True**, the service fills in “ciaddr” with 0.0.0.0 on reply (ACK) packets.

Num	Policy	Usage	Default Value	Description
24	Deny ConnectionList	`DenyConnectionList=IP addresses`	None	This policy does not allow connections from listed IP addresses and networks. An example of listed IP addresses would be: DenyConnectionList=127.0.0.1,10.0.0.0/8. In this example, connections from the loopback address and the Class A 10 network are not allowed. If this policy is set to **All**, connections from all IP addresses and networks are not allowed. If AllowConnectionList and DenyConnectionList are set at the same time, AllowConnectionList takes precedence over the DenyConnectionList.
25	AllowConnectionList	`AllowConnectionList=IP addresses`	All	This policy allows connections from all listed IP addresses and networks. An example of listed IP addresses would be: AllowConnectionList=127.0.0.1,10.0.0.0/8. In this example, connections from the loopback address and the Class A 10 network are allowed. If this policy is set to **All,** connections from all IP addresses and networks are allowed. If AllowConnectionList and DenyConnectionList are set at the same time. AllowConnectionList takes precedence over the DenyConnectionList.
26	ListenPort	`Any valid port number`	Ephemeral	This policy specifies which port the service listens for messages. Ephemeral indicates that the service will use a port that is dynamically allocated by the operating system. It will register this port with the local message service. To accept messages from previous releases of VitalQIP, set this policy to the service name qip_ctl, or the port number 1096. Ports are usually less than 32,000.
27	ACKPeriod	`Numeric`	0	This policy specifies how often an ACK will be expected when leases are transmitted to the GUI. By default, only the last packet is ACKed.

25 Configuring VRRP

The Virtual Router Redundancy Protocol is a standard router redundancy protocol supported in IPv4 and IPv6, based on RFC 3768 for VRRP version 2 (IPv4) and RFC 5798 for VRRP version 3 (IPv4 and IPv6). It provides redundancy by eliminating the single point of failure inherent in a default route environment. The VRRP router that controls the IPv4 or IPv6 address associated with a virtual router is called the master router and is responsible for forwarding virtual router advertisements. If the master router becomes unavailable, the highest priority backup router will transition to the master state. The OmniSwitch implementation of VRRP also supports the collective management of virtual routers on a switch.

Notes.

- Throughout the rest of this chapter, the feature name, "VRRP", refers to both IPv4 and IPv6 VRRP. However, "IPv4 VRRP" and "IPv6 VRRP" are used explicitly to highlight any differences.
- RFC 3768, which obsoletes RFC 2338, does not include support for authentication types. As a result, configuring VRRP authentication is no longer supported in this release.

In This Chapter

This chapter describes VRRP and how to configure it through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

This chapter provides an overview of VRRP and includes information about the following:

- "VRRP Defaults" on page 25-2.
- "Quick Steps for Creating a Virtual Router" on page 25-4.
- "VRRP Overview" on page 25-5.
- "Interaction With Other Features" on page 25-8.
- "Creating/Deleting a Virtual Router" on page 25-11.
- "Specifying an IP Address for a Virtual Router" on page 25-13.
- "Configuring the Advertisement Interval" on page 25-14.
- "Configuring Virtual Router Priority" on page 25-15.
- "Setting Preemption for Virtual Routers" on page 25-16.
- "Setting VRRP Traps" on page 25-18.
- "Configuring Collective Management Functionality" on page 25-19.
- "Creating VRRP Tracking Policies" on page 25-23.

- "Verifying the VRRP Configuration" on page 25-25.
- "IPv4 VRRP Application Example" on page 25-26.
- "IPv6 VRRP Application Example" on page 25-29.

VRRP Defaults

The following table lists the defaults for VRRP configuration through the **vrrp** command and the relevant command keywords:

Description	Keyword	Default
Virtual router enabled or disabled	**enable | disable**	Virtual routers are disabled
Priority	**priority**	100
Preempt mode	**preempt | no preempt**	Preempt mode is enabled
Accept mode	**accept | no accept**	Accept mode is enabled.
Advertising interval	**advertising interval**	100 centiseconds
VRRP version	**v2 | v3**	Version 2 for IPv4 virtual routers. Version 3 for IPv6 virtual routers.

The following table lists the defaults for VRRP configuration using the VRRP collective management features and the relevant command:

Default advertising interval for all the virtual routers on the switch.	**vrrp interval**	100 centiseconds
Default priority value for all the virtual routers on the switch.	**vrrp priority**	100
Default preempt mode for all the virtual routers on the switch.	**vrrp preempt**	**preempt**
Default accept mode for all the virtual routers on the switch.	**vrrp accept**	**accept**
Default version for all the virtual routers on the switch.	**vrrp version**	**v2** (for IPv4 virtual routers) **v3** (for IPv6 virtual routers)
Parameter value that is to be set and/or override with the new default value in all the virtual routers on the switch.	**vrrp set**	**all**
Default advertising interval for all the virtual routers in the group.	**vrrp group interval**	100 centiseconds
Default priority value for all the virtual routers in the group.	**vrrp group priority**	100
Default preempt mode for all the virtual routers in the group.	**vrrp group preempt | no preempt**	**preempt**

Default accept mode for all the virtual routers in the group.	**vrrp group accept | no accept**	**accept**
Default VRRP version for all the virtual routers in the group.	**vrrp group version v2 | v3**	**v2 (**for IPv4 virtual routers) **v3** (for IPv6 virtual routers)
Parameter value that is to be set and/or override with the new default value in all the virtual routers in the group.	**vrrp group set**	**all**

In addition, other defaults for VRRP include:

Description	Command	Default
VRRP delay	**vrrp delay**	45 seconds

Quick Steps for Creating a Virtual Router

1 Create a virtual router. Specify a virtual router ID (VRID) and an existing IPv4 or IPv6 interface name. For example, the following commands create an IPv4 and an IPv6 virtual router:

```
-> ip vrrp 23 interface ipv4-100
-> ipv6 vrrp 33 interface ipv6-200
```

For information about creating an IP interface, see Chapter 17, "Configuring IP," or Chapter 19, "Configuring IPv6."

2 Configure an IPv4 or IPv6 address for the virtual router.

```
-> ip vrrp 23 interface ipv4-100 address 192.168.173.1
-> ipv6 vrrp 33 interface ipv6-200 address 213:100:1::56
```

Note. An IPv4 address must be configured for an IPv4 virtual router before the virtual router can be administratively enabled. (This step is not required to administratively enable an IPv6 virtual router.)

3 Repeat steps 1 through 2 on all of the physical switches that will participate in backing up the address(es) associated with the virtual router.

4 Enable VRRP on each switch.

```
-> ip vrrp 23 interface ipv4-100 admin-state enable
-> ipv6 vrrp 33 interface ipv6-200 admin-state enable
```

Note. *Optional*. To verify the VRRP configuration, enter the **show vrrp** command.The display is similar to the one shown here:

```
-> show ip vrrp
VRRP default advertisement interval: 100 centiseconds
VRRP default priority: 100
VRRP default preempt: Yes
VRRP default accept: Yes
VRRP default version: V2
VRRP startup delay: 45  (expired)
VRRP BFD-STATE : Disabled

       Interface    IPv4                Admin                                Adv.
VRID     Name     Address(es)   Version Status   Priority Preempt Accept Interval
----+-----------+------------+-------+--------+--------+-------+------+---------
  23 ipv4-100     192.60.170.2    V2   Disabled      100     Yes     NA      100

-> show ipv6 vrrp
VRRP default advertisement interval: 100 centiseconds
VRRP default priority: 100
VRRP default preempt: Yes
VRRP default accept: Yes
VRRP startup delay: 50  (expired)
VRRP BFD-STATE : Disabled

       Interface                        Admin                                Adv.
VRID     Name      IPv6 Address(es      Status   Priority Preempt Accept Interval
----+-----------+-------------------+--------+---------+-------+------+--------
  33 ipv6-200      213:100:1::56      Enabled        100     Yes    Yes      100
```

For more information about this display, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

VRRP Overview

VRRP allows the routers on a LAN to backup a default route. VRRP dynamically assigns responsibility for a virtual router to a physical router (VRRP router) on the LAN. The virtual router is associated with an IP address (or set of IP addresses) on the LAN. A virtual router master is elected to forward packets for the virtual router's IP address. If the master router becomes unavailable, the highest priority backup router will transition to the master state.

Note. The IP address that is backed up may be the IP address of a physical router, or it may be a virtual IP address.

The example provided here is intended for understanding VRRP and does not show a configuration that would be used in an actual network.

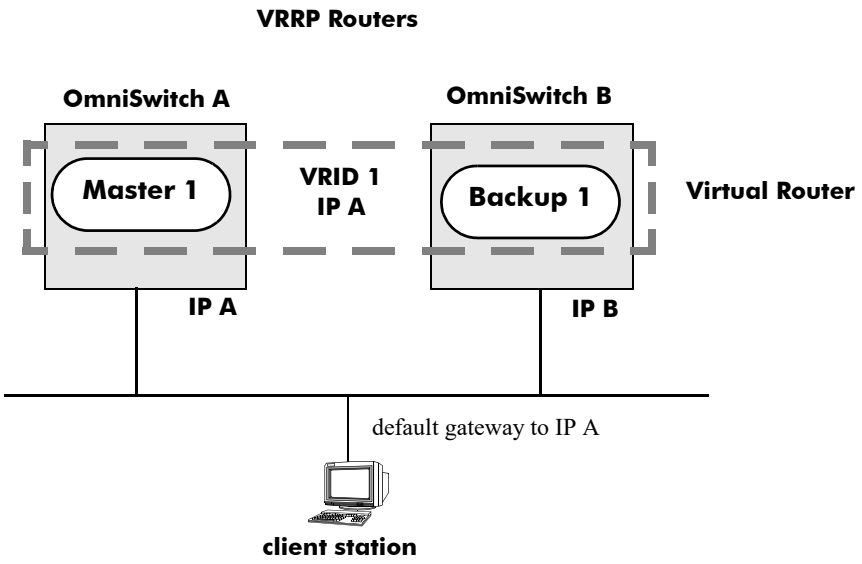


Figure 25-1 : VRRP Redundancy Example

In this example, each physical router is configured with a virtual router, VRID 1 which is associated with IP address A. OmniSwitch A is the master router because it contains the physical interface to which IP address A is assigned. OmniSwitch B is the backup router. The client is configured with a gateway address of IP A.

When VRRP is configured on these switches, and both the switches are available, OmniSwitch A will respond to ARP requests for IP address A using the virtual router's MAC address (00:00:5E:00:01:01 for version 2, 00:00:5E:00:02:01 for version 3) instead of the physical MAC address assigned to the interface. OmniSwitch A will accept packets sent to the virtual MAC address and forward them as appropriate; it will also accept packets addressed to IP address A (such as ping requests).

Note: A ping request to the VRRP IP will not be replied to if the request is from the local CMM which is also acting as the VRRP master. Only ping requests which originate from external routers will be replied to.

OmniSwitch B will respond to ARP requests for IP address B using the interface's physical MAC address. It will not respond to ARP requests for IP address A or to the virtual router MAC address.

If OmniSwitch A becomes unavailable, OmniSwitch B becomes the master router. OmniSwitch B will then respond to ARP requests for IP address A using the virtual router's MAC address (00:00:5E:00:01:01 for version 2, 00:00:5E:00:02:01 for version 3). It will also forward packets for IP address B and respond to ARP requests for IP address B using the OmniSwitch's physical MAC address.

OmniSwitch B uses IP address B to access the LAN. However, IP address B is not backed up. Therefore, when OmniSwitch B becomes unavailable, IP address B also becomes unavailable.

Why Use VRRP?

An end host may use dynamic routing or router discovery protocols to determine its first hop toward a particular IP destination. With dynamic routing, large timer values are required and may cause significant delay in the detection of a dead neighbor.

If an end host uses a static route to its default gateway, this creates a single point of failure if the route becomes unavailable. End hosts will not be able to detect alternate paths.

In either case, VRRP ensures that an alternate path is always available.

Definition of a Virtual Router

To backup an IP address or addresses using VRRP, a virtual router must be configured on VRRP routers on a common LAN. A VRRP router is a physical router running VRRP. A virtual router is defined by a virtual router identifier (VRID) and a set of associated IP addresses on the LAN.

Note. A single VRID may be associated with an IP interface that is bound to a VLAN or a Shortest Path Bridging (SPB) service.

Each VRRP router may backup one or more virtual routers. The VRRP router containing the physical interfaces to which the virtual router IP addresses are assigned is called the *IP address owner*. If it is available, the IP address owner will function as the master router. The master router assumes the responsibility of forwarding packets sent to the IP addresses associated with the virtual router and answering ARP requests for these addresses.

To minimize network traffic, only the master router sends VRRP advertisements on the LAN. The IP address assigned to the physical interface on the current master router is used as the source address in VRRP advertisements. The advertisements communicate the priority and state of the master router associated with the VRID to all VRRP routers. The advertisements are IP multicast datagrams sent to the VRRP multicast address 224.0.0.18 (as determined by the Internet Assigned Numbers Authority).

If a master router becomes unavailable, it stops sending VRRP advertisements on the LAN. The backup routers know that the master is unavailable based on the following algorithm:

Master Down Interval = (3 * *Advertisement Interval*) + *Skew Time*

where *Advertisement Interval* is the time interval between VRRP advertisements, and *Skew Time* is calculated based on the VRRP router's priority value as follows:

Skew Time = (256 - *Priority*) / 256

If the backup routers are configured with priority values that are close in value, there may be a timing conflict, and the first backup to take over may not be the one with the highest priority; and a backup with a higher priority will then preempt the new master. The virtual router may be configured to prohibit any preemption attempts, except by the IP address owner. An IP address owner, if it is available, will always become master of any virtual router associated with its IP addresses.

Note. Duplicate IP address/MAC address messages may display when a backup takes over for a master, depending on the timing of the takeover and the configured advertisement interval. This is particularly true if more than one backup is configured.

VRRP MAC Addresses

Each virtual router has a single well-known MAC address, which is used as the source in all periodic VRRP advertisements sent by the master router, as the MAC address in ARP replies sent by IPv4 VRRP, and as the MAC address in neighbor advertisements sent by IPv6 VRRP (instead of the MAC address for the physical VRRP router).

The IPv4 version 2 VRRP address has the following format:

00-00-5E-00-01-[virtual router ID]

The IPv4 version 3 and IPv6 VRRP address has the following format:

00-00-5E-00-02-[virtual router ID]

ARP Requests (IPv4 Virtual Routers)

Each IP virtual router has a single well-known MAC address, which is used as the MAC address in ARP instead of a VRRP router's physical MAC address. When an end host sends an ARP request to the master router’s IP address, the master router responds to the ARP request using the virtual router MAC address. If a backup router takes over for the master, and an end host sends an ARP request, the backup will reply to the request using the virtual router MAC address.

Gratuitous ARP requests for the virtual router IP address or MAC address are broadcast when the OmniSwitch becomes the master router. For VRRP interfaces, gratuitous ARP requests are delayed at system boot until both the IP address and the virtual router MAC address are configured.

If an interface IP address is shared by a virtual router, the routing mechanism does not send a gratuitous ARP for the IP address (since the virtual router will send a gratuitous ARP). This prevents traffic from being forwarded to the router before the routing tables are stabilized.

Neighbor Discovery (IPv6 Virtual Routers)

IPv6 virtual routers operate in much the same way as IPv4 virtual routers except for the following:

- The IPv6 Neighbor Discovery (ND) protocol is used instead of ARP to determine link layer addresses.
- ND Router Advertisements are sent to detect and advertise the existence and availability of IPv6 virtual routers.
- Virtual router advertisements are sent with the IPv6 interface link local address instead of the virtual router’s primary address.

ICMP Redirects

ICMP redirects are not sent out over VRRP interfaces.

VRRP Startup Delay

When a virtual router reboots and becomes master, it may become master before its routing tables are populated. This could result in loss of connectivity to the router. To prevent the loss in connectivity, a delay is used to prevent the router from becoming master before the routing tables are stabilized.

The default startup delay value can be modified to allow more or less time for the router to stabilize its routing tables.

In addition to the startup delay, the switch has an ARP delay (which is not configurable).

VRRP Tracking

A virtual router's priority may be conditionally modified to prevent another router from taking over as master. Tracking policies are used to conditionally modify the priority setting whenever a slot/port, remote IP address and/or IP interface associated with a virtual router goes down.

A tracking policy consists of a tracking ID, the value used to decrease the priority value, and the slot/port number, IP address, or IP interface name to be monitored by the policy. The policy is then associated with one or more virtual routers.

Configuring Collective Management Functionality

This feature provides user with the flexibility to manage the virtual routers on the switch collectively and also the capability to group the virtual routers to a virtual router group which simplifies the configuration and management tasks.

You can change the default values of the parameters like advertising interval, priority, preempt mode and the administrative status of all the virtual routers on a switch or in a virtual router group using this collective management functionality feature. For more information about configuring collective management functionality, see page 25-19.

Interaction With Other Features

- IP routing—IP routing must be enabled for the VRRP configuration to take effect.
- Router Discovery Protocol (RDP)—If RDP is enabled on the switch, and VRRP is enabled, RDP will advertise IP addresses of virtual routers depending on whether there are virtual routers active on the LAN, and whether those routers are backups or masters. When there are no virtual routers active on the LAN (either acting as master or backup), RDP will advertise all IP addresses. However, if virtual routers are active, RDP will advertise IP addresses for any master routers; RDP will not advertise IP addresses for backup routers.

IPv4 and IPv6 Interfaces

A VRRP virtual router instance is configured on an IPv4 or an IPv6 interface.

- The IP interface must exist and be bound to a VLAN or Shortest Path Bridging (SPB) service.
- If the IP interface is not bound to a VLAN or SPB service, the VRRP virtual router is not created.

- If the IP interface configured for the VRID is bound to an SPB service, then VRRP can operate over an SPB L3 VPN in-line routing configuration. See the “IP over SPBM” section of the “Configuring Shortest Path Bridging” chapter in this guide for more information about L3 VPN in-line routing.
- The IP addresses configured for a virtual router must be within the subnet of the IP interface address configured for the VRID. For example, if the address for IP interface “ipv4-100” is 10.10.2.1, then any addresses configured for the virtual router must be within that same subnet (such as 10.11.2.150, 10.12.3.254).

VRRP over UNP Dynamic SPB SAPs

When a dynamic UNP SAP connects two VRRP routers over an SPB backbone service, VRRP advertisements are sent through the SPB service domain to elect one router as the master and one as the slave (backup router). The slave router does not send out VRRP advertisements; only listens for advertisements from the master router. This inactivity may cause the dynamic UNP SAP on which the slave router communicates to age out. When this occurs, the slave router will no longer receive advertisements from the master router and will elect itself as the master. This results in two dual VRRP master routers operating within the same service domain.

To support VRRP communication over UNP SAP connections, the following UNP configuration is required *before* the VRRP routers are configured:

- Enable MAC address mobility for the UNP profile that is mapped to SPB service parameters. This will provide support for VRRP MAC address movement that is required for the VRRP master/slave election process.
- Statically assign the UNP service profile to a UNP access port to create a persistent UNP SAP on which a VRRP router will communicate. A persistent SAP does not age out and will ensure an uninterrupted flow of VRRP advertisements between the VRRP master and slave routers.

See “Configuring UNP Profile Attributes” and “Statically Assigning Service Profiles for Silent Devices” in the “Configuring Access Guardian” chapter of this guide for more information about enabling MAC mobility and creating a persistent SAP.

VRRP Tracking with BFD

When VRRP tracking is used to monitor a remote IP address, the virtual router’s priority adjusts based on the reachability of the remote address. If the virtual router doing the tracking (the master) cannot reach the remote address, the master gives up control and lets a router that can reach the remote address takeover.

To facilitate a sub-second takeover from the master in this scenario, configure a VRRP address tracking policy on the slave for the IP interface address of the master virtual router, set the policy priority to zero, and enable Bidirectional Forwarding Detection (BFD) for the policy.

Notes:

- Setting the policy priority to zero is what signals the slave router to become master. If the priority value of the slave tracking policy is *not* set to zero, the tracking policy is treated normally and the priority of the slave is reduced when the remote address becomes unreachable.
- Configuring the Loopback0 IP interface on the switch is required when enabling BFD for a remote address tracking policy. The IP network address assigned to Loopback0 is used as the source IP address for BFD packets.

Virtual Routing and Forwarding (VRF)

- VRRP runs as a separate task within each VRF instance. When a max profile VRF is created, the VRRP task automatically starts. For low profile VRFs, the VRRP task is started with the **ip load vrrp** command
- An IPv6 VRID number must be unique across all VRF instances.
- Tracking policies are defined and managed within the context of a VRF instance and on a per-VRF basis, except for port tracking policies. Switch ports are not bound to a specific VRF instance. Port-tracking policies defined within any VRF instance are applied to all switch ports across all VRF instances.
  - An interface tracking policy is applied only to interfaces defined within the same VRF instance as the tracking policy.
  - An address tracking policy is applied only to addresses that are reachable via interfaces defined within the same VRF instance as the tracking policy.
- Virtual router groups are defined and managed within the context of a specific VRF instance. A virtual router group can only contain VRIDs defined within the same VRF instance as the router group.

VRRP Configuration Overview

During startup, VRRP is loaded onto the switch and is enabled. Virtual routers must be configured and enabled as described in the following sections. Since VRRP is implemented on multiple switches in the network, some VRRP parameters must be identical across switches:

- **VRRP and ACLs**

  If QoS filtering rules (Access Control Lists) are configured for Layer 3 traffic on a VRRP router, all of the VRRP routers on the LAN must be configured with the same filtering rules; otherwise the security of the network will be compromised. See Chapter 28, "Configuring QoS," for more information about filtering.

- **Conflicting VRRP Parameters Across Switches**

  All virtual routers with the same VRID on the LAN should be configured with the same advertisement interval and IP addresses. If the virtual routers are configured differently, it may result in more than one virtual router acting as the master router. This in turn would result in duplicate IP and MAC address messages as well as multiple routers forwarding duplicate packets to the virtual router MAC address. Use the **show vrrp** command to check for conflicting parameters. For information about configuring VRRP parameters, see the remaining sections of this chapter.

Basic Virtual Router Configuration

At least two virtual routers must be configured on the LAN—a master router and a backup router. The virtual router is identified by a number called the Virtual Router ID (VRID), the IP interface name on which the virtual router is configured, and the IP address or addresses associated with the router. Multiple virtual routers may be configured on a single physical VRRP router.

Basic commands for setting up virtual routers include:

vrrp
vrrp address

The next sections describe how to use these commands.

Creating/Deleting a Virtual Router

There are two **vrrp** command options for creating a virtual router:

- **ip vrrp**—creates a VRRP virtual router for IPv4 addresses.
- **ipv6 vrrp**—creates a VRRP virtual router for IPv6 addresses.

Both of these command options require a VRID and an existing IPv4 or IPv6 interface to create the virtual router. The VRID must be a unique number ranging from 1 to 255.

For example, the following commands create an IPv4 VRRP virtual router with VRID 6 on interface "ipv4-100" and an IPv6 VRRP virtual router with VRID 10 on interface "ipv6-200":

```
-> ip vrrp 6 interface ipv4-100
-> ipv6 vrrp 10 interface ipv6-200
```

In addition to the required VRID and IP interface name, the following optional parameters may also be specified when creating an IPv4 or IPv6 VRRP virtual router:

- ***Priority***: Use the **priority** keyword to change the default priority value and set a desired value. Note that the IP address owner is automatically assigned a value of 255, which overrides any value that you may have already configured. See "Configuring Virtual Router Priority" on page 25-15 for more information about how priority is used.
- ***Preempt mode***: To change from the default preempt mode and to turn it off, use **no preempt**. Use **preempt** to turn it back on. For more information about the preempt mode, see "Setting Preemption for Virtual Routers" on page 25-16.
- ***Accept mode***: The **accept** mode applies only to IPv4 version 3 and IPv6 virtual routers and allows the master router to accept packets addressed to the IP address owner as its own. Use the **no accept** mode to prevent the master router from accepting packets addressed to the IP address owner. See "Setting the Accept Mode" on page 25-16.
- ***Advertising interval***: Measured in centiseconds. Use the **interval** keyword with the desired number of centiseconds for the delay in sending VRRP advertisement packets. You can change the default interval value and set a desired value. See "Configuring the Advertisement Interval" on page 25-14.

  Notes.

  - The maximum number of VRRP virtual routers supported is based on the 100 centisecond interval. A smaller interval will result in a relatively lesser number of virtual routers.
  - The centisecond interval cannot be less than 10 centiseconds and must increment by 10 (for example, 10, 20, 30) up to the maximum allowed.

- ***VRRP version***: To change the VRRP version for IPv4 virtual routers, use **version v2** or **version v3**. The VRRP version for IPv6 virtual routers is not configurable (IPv6 virtual routers only support version 3). See "Configuring the VRRP Version" on page 25-17.

The following example creates an IPv4 VRRP virtual router (with VRID 7) on interface "ipv4-100" with a priority of 75. The preempt mode of the router is enabled and VRRP advertisements will be sent at intervals of 2 seconds:

```
-> ip vrrp 7 interface ipv4-100 priority 75 preempt interval 200
```

The following example creates an IPv6 virtual router (with VRID 11) on interface "ipv6-200" with a priority of 75, no preempt, and no accept. VRRP advertisements will be sent at intervals of 200 centiseconds:

```
-> ipv6 vrrp 11 interface ipv6-200 priority 75 no preempt no accept interval 200
```

Note. All virtual routers with the same VRID on the same LAN should be configured with the same advertising interval; otherwise the network may produce duplicate IP or MAC address messages.

The **vrrp** command may also be used to specify whether the virtual router is administratively enabled or disabled. *However, an IPv4 virtual router must have an IP address assigned to it before it can be enabled (not required to enable an IPv6 virtual router)*. Use the **vrrp address** command as described in the next section to specify an IP address or addresses.

To delete a virtual router, use the **no** form of the **ip vrrp** or **ipv6 vrrp** command option with the relevant VRID and IP interface. For example:

```
-> no ip vrrp 7 interface ipv4-100
-> no ipv6 vrrp 11 interface ipv6-200
```

Virtual router 7 on interface "ipv4-100" and virtual router 11 on interface "ipv6-200" are deleted from the configuration. (The virtual router does not have to be disabled before you delete it.)

For more information about the **vrrp** command syntax, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Specifying an IP Address for a Virtual Router

To assign an IP address to an IPv4 or IPv6 virtual router, use one of the following **vrrp address** command options:

- **ip vrrp address**—specifies an IPv4 address for an IPv4 VRRP virtual router.
- **ipv6 vrrp address**—specifies an IPv6 address for an IPv6 VRRP virtual router.

A VRRP virtual router must have a link local address. By default, the virtual router link local address is created based on the virtual router MAC address and it does not need to be configured.

An IP address must be configured before an IPv4 virtual router can be administratively enabled. This is not required to administratively enable an IPv6 virtual router. In addition, IP addresses configured for an IPv4 or IPv6 VRRP virtual router must be within the subnet of the IP interface address configured for the virtual router. For example, the following commands specify an IPv4 address for VRID 6 that is within the subnet of the address configured for interface “ipv4-100” and an IPv6 address for VRID 10 that is within the subnet of the address configured for interface “ipv6-200”.

```
-> ip interface ipv4-100 address 10.10.2.1 vlan 100
-> ip vrrp 6 interface ipv4-100 address 10.10.2.5
-> ip vrrp 6 interface ipv6-200 admin-state enable

-> ipv6 interface ipv6-200 address 213:100::50 vlan 200
-> ipv6 vrrp 10 interface ipv6-200 address 213:100:1::56
-> ipv6 vrrp 10 interface ipv6-200 admin-state enable
```

In this example, the VRRP virtual routers are administratively enabled after the IP address is assigned.

Note that if a virtual router is to be the IP address owner, then all addresses on the virtual router must match an address on the switch interface. This includes the IPv6 virtual router's link local address. In other words, a virtual router can not be the IPv6 address owner if its link local address does not match the interface link local address.

To remove an IP address from a virtual router, use the **no** form of the **ip vrrp address** or **ipv6 vrrp address** command option. For example:

```
-> ip vrrp 6 interface ipv4-100 admin-state disable
-> ip vrrp 6 interface ipv4-100 no address 10.10.2.3

-> ipv6 vrrp 10 interface ipv6-200 admin-state disable
-> ipv6 vrrp 10 interface ipv6-200 no address 213:100:1::56
```

In this example, IPv6 virtual router 6 and IPv6 virtual router 10 are disabled. (A virtual router must be disabled before IP addresses may be added or removed from the router.)

IPv4 address 10.10.2.3 is then removed from the virtual router with the **no** form of the **ip vrrp address** command, and IPv6 address 213:100:1::56 is then removed from the virtual router with the **no** form of the **ipv6 vrrp address** command.

Configuring the Advertisement Interval

Consider the following when configuring the advertisement interval value:

- IPv4 version 2 virtual routers with the same VRID must be configured to use the same interval value. If this value is set differently for a master router and a backup router, the IPv4 version 2 virtual router behavior is as follows:
  - IPv4 VRRP packets may get dropped because the newly configured interval does not match the interval indicated in the packet.
  - The backup virtual router will then take over and send a gratuitous ARP, which includes the virtual router IPv4 address and the virtual router MAC address. In addition to creating duplicate IPv4/MAC address messages, both routers will begin forwarding packets sent to the virtual router MAC address. This will result in the forwarding of duplicate packets.
- For IPv4 version 3 or IPv6 virtual routers, the backup router will adapt to the advertising interval set for the master router; there is no mismatch and the configured value for the backup router is not changed. When the backup router becomes master, then advertisements are sent using the backup's configured advertising interval value.
- The VRRP router version determines the advertising interval value range and the level of granularity that is used to specify the interval value. For example:
  - The advertising interval range for VRRP version 2 is 1–25500 and cannot be less than 100 centiseconds. The interval value should also be entered in increments of 100 (for example, 100, 200, 300) up to the maximum allowed. If an interval value is not specified in an increment of 100, then the value is rounded down to the next increment of 100. For example, if a value of 209 is specified, then that value is rounded down to 200.
  - The advertising interval range for VRRP version 3 is 1–4095 and cannot be less than 10 centiseconds. The interval value should also be entered in increments of 10 (for example, 10, 20, 30) up to the maximum allowed. If the interval value is not specified in an increment of 10, then the value is rounded down to the next increment of 10. For example, if a value of 15 is specified, then that value is rounded down to 10.

To configure the advertisement interval, use the **ip vrrp** or **ipv6 vrrp** command option with the **interval** keyword. For example:

```
-> ip vrrp 6 interface ipv4-100 admin-state disable
-> ip vrrp 6 interface ipv4-100 interval 200

-> ipv6 vrrp 10 interface ipv6-200 admin-state disable
-> ipv6 vrrp 10 interface ipv6-200 interval 500
```

In this example, IPv4 virtual router 6 and IPv6 virtual router 10 are disabled. (When modifying an existing virtual router, the virtual router must be disabled before it can be modified.) The **ip vrrp** command is then used to set the advertising interval for virtual router 6 to 200 centiseconds, and the **ipv6 vrrp** command is then used to set the advertising interval for virtual router 10 to 500 centiseconds.

The following example shows how the specified interval value of 250 is automatically rounded down to the next increment of 100 for a version 2 VRRP router:

```
-> ip vrrp 20 interface ipv4-100 admin-state disable
-> ip vrrp 20 interface ipv4-100 interval 250

-> show ip vrrp 20
Virtual Router VRID = 20 on INTERFACE = ipv4-100
  Version       = V2
  Admin. Status = Disabled
  Priority      = 100
  Preempt       = Yes
```

```
  Adv. Interval = 200
  Virtual MAC   = 00-00-5E-00-01-0A
  IP Address(es)
    192.60.170.2
    192.60.170.3
```

The following example shows how the specified interval value of 133 is automatically rounded down to the next increment of 10 for a version 3 VRRP router:

```
-> ipv6 vrrp 33 interface v6if-v200 admin-state disable
-> ipv6 vrrp 33 interface v6if-v200 interval 133

-> show ipv6 vrrp 33
Virtual Router VRID = 33 on INTERFACE = v6if-v200
  Version       = V3
  Admin. Status = Enabled
  Priority      = 100
  Preempt       = Yes
  Accept        = Yes
  Adv. Interval = 130
  Virtual MAC   = 00-00-5E-00-02-21
  IP Address(es)
    fe80::200:5eff:fe00:221
    2001:db8:4100:1::1
```

Configuring Virtual Router Priority

VRRP functions with one master virtual router and at least one backup virtual router. A priority value determines the role each router plays. It also decides the selection of backup routers for taking over as the master router, if the master router is unavailable.

Priority values range from 1 to 255. A value of 255 indicates that the virtual router owns the IP address; that is, the router contains the real physical interface to which the IP address is assigned. The switch can change the default value and set it to 255 if it detects that the router is the IP address owner. The value cannot be set to 255 if the router is not the IP address owner.

The IP address owner will always be the master router if it is available. If more than one backup router is configured, their priority values should be configured with different values, so that the backup with the higher value will take over for the master. The priority parameter may be used to control the order in which backup routers will take over for the master. If priority values are the same, the master router is selected as follows:

- IPv4 VRRP selects the backup virtual router with the highest physical interface IPv4 address to take over for the master.
- IPv6 VRRP will select any backup virtual router to take over for the master.
  - Note that the switch sets the priority value to zero in the last IPv6 VRRP advertisement packet before a master router is disabled.
  - If a router is the IPv6 address owner and the priority value is not set to 255, the switch will set its priority to 255 when the router is enabled.

To set the priority, use the **ip vrrp** or **ipv6 vrrp** command option with the **priority** keyword and the desired value. For example:

```
-> ip vrrp 6 interface ipv4-100 admin-state disable
-> ip vrrp 6 interface ipv4-100 priority 50
```

```
-> ipv6 vrrp 10 interface ipv6-200 admin-state disable
-> ipv6 vrrp 10 interface ipv6-200 priority 50
```

In this example, IPv4 virtual router 6 and IPv6 virtual router 10 are disabled. (If you are modifying an existing virtual router, the virtual router must be disabled before it can be modified.) The virtual router priority is then set to 50. The priority value is relative to the priority value configured for other virtual routers backing up the same IPv4 or IPv6 address. Setting the priority value to 50 would typically provide a router with a lower priority in the IPv4 or IPv6 VRRP network.

Setting Preemption for Virtual Routers

When a master virtual router becomes unavailable (goes down for whatever reason), a backup router will take over. When there is more than one backup router and if their priority values are very nearly equal, the skew time may not be sufficient to overcome delays caused by network traffic loads. This may cause a lower priority backup to assume control before a higher priority backup. But when the preempt mode is enabled, the higher priority backup router will detect this and assume control.

Note. In certain cases, this may not be a desirable behavior, as when the original master comes back and immediately causes all the traffic to switch back to it.

If all virtual routers have the preempt mode enabled, the virtual router with the highest priority will become the master. If the master router goes down, the highest priority backup router will become the master. If the previous master or any other virtual router comes up with the preempt mode enabled and has a higher priority value, this router will become the new master.

By default virtual routers are allowed to preempt each other; that is, the virtual router with the highest priority will take over if the master router becomes unavailable. The preempt mode may be disabled so that any backup router that takes over when the master is unavailable will not then be preempted by a backup with a higher priority.

Note. The virtual router that owns the IP address(es) associated with the physical router always becomes the master router if is available, regardless of the preempt mode setting and the priority values of the backup routers.

To prevent a router with a higher priority value from automatically taking control from a master router with a lower priority value, disable the preempt mode for the higher priority router. This is done by using the **no preempt** keywords with the **ip vrrp** or **ipv6 vrrp** command option. For example:

```
-> ip vrrp 6 interface ipv4-100 admin-state disable
-> ip vrrp 6 interface ipv4-100 no preempt
-> ipv6 vrrp 10 interface ipv6-200 admin-state disable
-> ipv6 vrrp 10 interface ipv6-200 no preempt
```

In this example, IPv4 virtual router 6 and IPv6 virtual router 10 are disabled. (If you are modifying an existing virtual router, the virtual router must be disabled before it may be modified.) The preempt mode is then disabled for the virtual routers. If virtual router 6 and 10 take over for an unavailable router, another router with a higher priority will not be able to preempt them. For more information about priority, see "Configuring Virtual Router Priority" on page 25-15.

Setting the Accept Mode

The accept mode specifies whether a master virtual router will accept packets addressed to a virtual IP address when the router itself is not the owner of that IP address. When a master virtual router receives a packet destined for a virtual IP address, the packet is dropped unless one of the following is true:

- The virtual router receiving the packet is the owner of the IP address.
- The IPv4 version 3 router or IPv6 router receiving the packet is not the owner of the IP address but the accept mode is set. The packet is processed and replied to by the router.

When an IPv4 version 2 router receives a packet destined for a virtual IP address and the router is not the owner of the IP address, the packet is not accepted. The accept mode is not configurable for an IPv4 version 2 router, however, the router will still respond to pings sent to the destined virtual IP address.

By default, the accept mode is set. To change the mode, use the **no accept** keyword with the **ip vrrp** or **ipv6 vrrp** command. For example:

```
-> ip vrrp 6 interface ipv4-100 admin-state disable
-> ip vrrp 6 interface ipv4-100 no accept

-> ipv6 vrrp 10 interface ipv6-200 admin-state disable
-> ipv6 vrrp 10 interface ipv6-200 no accept
```

In this example, IPv4 virtual router 6 and IPv6 virtual router 10 are disabled. (If you are modifying an existing virtual router, the virtual router must be disabled before it may be modified.) The accept mode is then disabled for the virtual routers.

Configuring the VRRP Version

VRRP version 2 supports IPv4 virtual routers; VRRP version 3 supports both IPv4 and IPv6 virtual routers. By default, IPv4 virtual routers are configured to use VRRP version 2. To select version 3 for an IPv4 virtual router, use the **version v3** keyword with the **ip vrrp** command. For example:

```
-> ip vrrp 6 interface ipv4-100 version v3
```

IPv6 virtual routers always use VRRP version 3 (only version 3 supports IPv6). As a result, the version number is not configurable for IPv6 virtual routers.

Note. There is no interoperability between VRRP version 2 and 3. A version 3 virtual router will not acknowledge version 2 advertisements.

Enabling/Disabling a Virtual Router

To administratively enable a virtual router, use the **ip vrrp** or **ipv6 vrrp** command option with the **admin-state enable** keyword. Note that at least one IPv4 address must be configured for an IPv4 virtual router before the virtual router can be enabled; this is not a requirement for IPv6 virtual routers. For example:

```
-> ip vrrp 6 interface ipv4-100 priority 150
-> ip vrrp 6 interface ipv4-100 address 10.10.2.3
-> ip vrrp 6 interface ipv4-100 admin-state enable

-> ipv6 vrrp 10 interface ipv6-200 priority 150
-> ipv6 vrrp 10 interface ipv6-200 address 213:100:1::56
-> ipv6 vrrp 10 interface ipv6-200 admin-state enable
```

In this example, an IPv4 virtual router is created on interface “ipv4-100” with a VRID of 6 and an IPv6 virtual router is created on interface “ipv6-200” with a VRID of 10. An IPv4 address is then assigned to the IPv4 virtual router and an IPv6 address is then assigned to the IPv6 virtual router. Both virtual routers are then enabled on the switch.

To disable a virtual router, use the **admin-state disable** keyword.

```
-> ip vrrp 6 interface ipv4-100 admin-state disable
-> ipv6 vrrp 10 interface ipv6-200 admin-state disable
```

A virtual router must be disabled before it can be modified. Use the **ip vrrp** or **ipv6 vrrp** command option to disable the virtual router first; then use the command again to modify the parameters. For example:

```
-> ip vrrp 6 interface ipv4-100 admin-state disable
-> ip vrrp 6 interface ipv4-100 priority 200
-> ip vrrp 6 interface ipv4-100 admin-state enable

-> ipv6 vrrp 10 interface ipv6-200 admin-state disable
-> ipv6 vrrp 10 interface ipv6-200 priority 200
-> ipv6 vrrp 10 interface ipv6-200 admin-state enable
```

In this example, IPv4 virtual router 6 on interface “ipv4-100” and IPv6 virtual router 10 on interface “ipv6-200” are disabled. Both virtual routers are then modified to change their priority setting. (For information about configuring the priority setting, see “Configuring Virtual Router Priority” on page 25-15.) After the priority settings are changed, the virtual routers are then re-enabled and will be active on the switch.

Setting VRRP Traps

A VRRP router has the capability to generate VRRP SNMP traps for events defined in the VRRP SNMP MIB. In order for VRRP traps to be generated correctly, traps in general must be enabled on the switch through the SNMP CLI. See the *OmniSwitch AOS Release 8 Switch Management Guide* for more information about enabling SNMP traps globally.

Setting VRRP Startup Delay

After a switch reboot, the global delay value takes effect and all virtual routers (IPv4 and IPv6) remain in the **initialize** state. They will remain in this state until the timer expires, at which point they will negotiate to determine whether to become the master or a backup.

To set a delay to keep all virtual routers from going active before their routing tables are set up, use the **vrrp delay** command. This command applies only when the switch reboots.

```
-> ip vrrp delay 75
```

The switch now waits 75 seconds after its reboot before it becomes available to take over as master for another router.

Notes:

- This command applies only when the switch reboots.
- All IPv4 and IPv6 virtual routers on the switch remain in the initialize state during the startup delay time that is set using the **ip vrrp delay** command; there is no separate command option for IPv6 virtual routers.

Configuring Collective Management Functionality

Collective management simplifies the management and configuration tasks of either all the IPv4 or IPv6 virtual routers on the switch or only the virtual routers in a particular virtual router group.

The following section describes the above mentioned collective management functionality in detail:

Changing Default Parameter Values for all Virtual Routers

You can change the default advertising interval value of all the IPv4 or IPv6 virtual routers on a switch using the **ip** or **ipv6** form of the **vrrp interval** command. For example:

```
-> ip vrrp interval 50
-> ipv6 vrrp interval 50
```

You can change the default priority value of all the IPv4 or IPv6 virtual routers on a switch using the **ip** or **ipv6** form of the **vrrp priority** command. For example:

```
-> ip vrrp priority 50
-> ipv6 vrrp priority 50
```

You can change the default preempt mode of all the IPv4 or IPv6 virtual routers on a switch using the **ip** or **ipv6** form of the **vrrp preempt** command. For example:

```
-> ip vrrp no preempt
-> ipv6 vrrp no preempt
```

You can change the default accept mode of all IPv4 version 3 and IPv6 virtual routers on a switch using the **ip** or **ipv6** form of the **vrrp accept** command. For example:

```
-> ip vrrp no accept
-> ipv6 vrrp no accept
```

You can change the default VRRP version of all the IPv4 virtual routers on a switch using the **vrrp version** command. For example:

```
-> ip vrrp version v3
```

These commands will set the new default values only for the virtual routers that are newly created. However, you can apply the new default value to the existing virtual routers. To apply the new default value to the existing virtual routers; you must first disable the virtual routers, then apply the new default value using the **ip** or **ipv6** form of the **vrrp set** command and enable the virtual routers again.

For example, to change the default priority value to 50 on all the existing IPv4 or IPv6 virtual routers on a switch, enter the following:

```
-> ip vrrp priority 50
-> ip vrrp admin-state disable
-> ip vrrp set priority
-> ip vrrp admin-state enable

-> ipv6 vrrp priority 50
-> ipv6 vrrp admin-state disable
-> ipv6 vrrp set priority
-> ipv6 vrrp admin-state enable
```

The first command configures the default priority value as 50 for all the virtual routers on the switch. The next command disables all the virtual routers on the switch. The **ip vrrp set** and **ipv6 vrrp set** command in each sequence applies the new default priority value to the existing virtual routers. This value will be

applied only to the virtual routers that already have the default value and not the values configured either individually or via group. This is because the configured values take priority over the default values.

For the modified default values to effect the virtual routers which are configured with a value either individually or via group, you can use the **ip** or **ipv6** form of the **vrrp set** command along with the **override** option. For example:

```
-> ip vrrp set priority override
-> ipv6 vrrp set priority override
```

Note. You can specify a parameter such as **interval**, **priority**, **preempt**, **accept**, **version**, or **all** with the **ip** or **ipv6** form of the **vrrp set** command to set and/or override the existing value with the new default values. The **all** option resets and/or overrides the existing advertising interval value, priority value, preempt mode, accept mode, and version with the modified default values.

The next command in the example enables all the virtual routers on the switch except the virtual routers that are disabled individually or via group. To enable all the virtual routers on the switch including those which are disabled individually or via group, you can use the **ip** or **ipv6** form of the **vrrp admin-state** command along with the **enable-all** option. For example:

```
-> ip vrrp admin-state enable-all
-> ipv6 vrrp admin-state enable-all
```

Note. This collective virtual routers management functionality will not affect the ability to change the administrative status and parameter values of an individual virtual router.

Changing Default Parameter Values for a Virtual Router Group

The virtual routers can also be grouped under a virtual router group as another way of simplifying the configuration and management tasks.

An IPv4 or IPv6 virtual router group can be created using the **ip** or **ipv6** form of the **vrrp group** command. For example:

```
-> ip vrrp group 25
-> ipv6 vrrp group 30
```

These commands create an IPv4 virtual router group 25 and an IPv6 virtual router group 30. Use the **no** form of the same commands to delete a virtual router group. For example:

```
-> no ip vrrp group 25
-> no ipv6 vrrp group 30
```

Note. When a virtual router group is deleted, the virtual routers assigned to the group become unassigned. However, this does not have any impact on the virtual routers.

After creating a virtual router group, you have to add virtual routers to the group using the **ip** or **ipv6** form of the **vrrp group-association** command. For example:

```
-> ip vrrp 6 interface ipv4-100 group-association 25
-> ipv6 vrrp 10 interface ipv6-200 group-association 30
```

The above commands add the IPv4 virtual router 6 on interface “ipv4-100” to the virtual router group 25 and the IPv6 virtual router 10 on interface “ipv6-200” to the virtual router group 30. A virtual router need not be disabled in order to be added to a virtual router group. However, the virtual router will not adopt the group’s default parameter values until those values are applied by re-enabling the virtual router.

To remove a virtual router from a virtual router group, use the **no** form of the same commands as follows:

```
-> ip vrrp 6 interface ipv4-100 no group-association 25
-> ipv6 vrrp 10 interface ipv6-200 no group-association 30
```

Note that a virtual router need not to be disabled to be removed from a group.

You can change the default values of the parameters (such as **interval**, **priority**, **preempt**, **accept**, and **version**) for all of the virtual routers in a virtual router group using the **ip** or **ipv6** form of the **vrrp group** command, as follows:

```
-> ip vrrp group 25 interval 50 priority 50 no preempt
-> ipv6 vrrp group 30 interval 50 priority 50 no preempt
```

The above command configures the default value for advertising interval as 50 seconds, priority as 150 and preempting mode as **no preempt**. These parameters can be modified at any time but will not have any effect on the virtual routers in the group until you disable, then apply the group default value using the **ip** or **ipv6** form of the **vrrp group set** command and enable the virtual router group again.

For the modified default values to be applied to the virtual routers in a group, you must disable the virtual router group, then apply the group default value using the **ip vrrp group set** or **ipv6 vrrp group set** command and enable the virtual router group again. For example:

```
-> ip vrrp group 25 interval 50
-> ip vrrp group 25 admin-state disable
-> ip vrrp group 25 set interval
-> ip vrrp group 25 admin-state enable


-> ipv6 vrrp group 30 interval 50
-> ipv6 vrrp group 30 admin-state disable
-> ipv6 vrrp group 30 set interval
-> ipv6 vrrp group 30 admin-state enable
```

The first command in each example configures the default interval value as 50 for all the virtual routers in the virtual router group (IPv4 group 25 and IPv6 group 30). The next command disables all the virtual routers in the group. The **ip vrrp group set** and **ipv6 vrrp group set** command in each example applies the new default interval value to all the virtual routers in the group. This value will be applied only to the virtual routers in the group that already have the default value and not the values configured individually. This is because the configured values take priority over the default values.

For the modified default values to affect the virtual routers in the group, including the virtual routers that are configured with a value individually, you can use the **ip** or **ipv6** form of the **vrrp group set** command along with the **override** option. For example:

```
-> ip vrrp group 25 set interval override
-> ipv6 vrrp group 30 set interval override
```

Note. You can specify a parameter such as **interval**, **priority**, **preempt**, **accept**, **version**, or **all** with the **ip** or **ipv6** form of the **vrrp group set** command to set and/or override the existing value with the new default values. The **all** option resets and/or overrides the existing advertising interval value, priority value, preempt mode, accept mode, and version with the modified default values.

The next command enables all the virtual routers in the group except the virtual routers that are disabled individually. To enable all the virtual routers in the group including those which are disabled individually, you can use the **ip** or **ipv6** form of the **vrrp group admin-state** command with the **enable-all** option as follows:

```
-> ip vrrp group 25 admin-state enable-all
-> ipv6 vrrp group 30 admin-state enable-all
```

Note. Even though a virtual router may be assigned to a group, its parameter values and administrative status can still be modified individually.

Creating VRRP Tracking Policies

To create a tracking policy, use the **vrrp track** command and specify the amount to decrease a virtual router's priority and the slot/port, IP address, or IP interface name to be tracked. For example:

```
-> ip vrrp track 3 admin-state enable priority 50 address 20.1.1.3
```

In this example, a tracking policy ID (3) is created and enabled for IP address 20.1.1.3. If this address becomes unreachable, a virtual router associated with this track ID will have its priority decremented by 50. Note that the **enable** keyword administratively activates the tracking policy, but the policy does not take effect until it is associated with one or more virtual routers (see the next section).

Similarly, to create a tracking policy ID (3) for IPv6 address 213:100:1::56, use the following command:

```
-> ip vrrp track 3 admin-state enable priority 50 address 213:100:1::56
```

If this address becomes unreachable, a virtual router associated with this track ID will have its priority decremented by 50.

When creating a policy to track a remote IP address, the following optional parameter settings are available to configure:

- ***BFD status***: The status of Bidirectional Forwarding Detection for IPv4 and IPv6 address tracking policies. Use the **bfd-state** parameter with the **ip vrrp track address** command to enable or disable the BFD status. See "VRRP Tracking with BFD" on page 25-9 for more information.
- ***Delay time***: Use the **delay** parameter with the **ip vrrp track address** command to specify the amount of time to wait after a VRRP address track is detected as operationally up and before the associated virtual router's priority value is incremented by the tracking policy's priority value. Setting a delay value can help to prevent the loss of device connectivity that may occur when a virtual router prematurely becomes the master.

Note the following:

- IPv4 and IPv6 tracking policies are created using the same **ip vrrp track** command; there isn't a separate command option for IPv6 policies.
- A virtual router must be administratively disabled before a tracking policy for the virtual router can be added.
- VRRP tracking does not override IP address ownership (the IP address owner will always have priority to become master, if it is available).

Associating a Tracking Policy with a Virtual Router

There are two **vrrp track-association** command options for associating a tracking policy with a virtual router:

- **ip vrrp track-association**—associates a tracking policy with an IPv4 virtual router.
- **ipv6 vrrp track association**—associates a tracking policy with an IPv6 virtual router.

To associate a tracking policy with an IPv4 or IPv6 virtual router, use the **ip vrrp track-association** or the **ipv6 vrrp track-association** command option with the tracking policy ID number. For example:

```
-> ip vrrp 6 interface ipv4-100 admin-state disable
-> ip vrrp 6 interface ipv4-100 track-association 3
```

```
-> ipv6 vrrp 10 interface ipv6-200 admin-state disable
-> ipv6 vrrp 10 interface ipv6-200 track-association 3
```

In this example, IPv4 virtual router 6 on interface “ipv4-100” and IPv6 virtual router 10 on interface “ipv6-200” are disabled first so that tracking policy 3 may be associated with the virtual router. When the virtual router is re-enabled, tracking policy 3 will be used for that virtual router.

A tracking policy should not be associated with a virtual router on the same port or interface. For example:

```
-> ip interface vlan-4 address 10.1.1.1 vlan 4
-> ip vrrp track 2 ipv4-interface vlan-4
-> ip vrrp 5 interface vlan-4 track-association 2
```

This configuration is allowed but will not really have an effect. If the associated interface goes down, this virtual router goes down as well and the tracking policy is not applied.

Note. A master and a backup virtual router should not be tracking the same IP address; otherwise, when the IP address becomes unreachable, both virtual routers will have their priorities decremented, and the backup may temporarily take over if the master discovers that the IP address is unreachable before the backup.

Typically you should not configure the same IP address tracking policies on physical VRRP routers that backup each other; otherwise, the priority will be decremented for both master and backup when the entity being tracked goes down.

Verifying the VRRP Configuration

A summary of the **show** commands used for verifying the VRRP configuration is given here:

show vrrp	Displays the virtual router configuration for all virtual routers or for a particular virtual router.
show vrrp statistics	Displays statistics about VRRP packets for all virtual routers configured on the switch or for a particular virtual router.
show vrrp track	Displays information about tracking policies on the switch.
show vrrp track-association	Displays the tracking policies associated with virtual routers.
show vrrp group	Displays the default parameter values for all the virtual router groups or for a specific virtual router group.
show vrrp group-association	Displays the virtual routers that are associated with a group.

For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

IPv4 VRRP Application Example

In addition to providing redundancy, IPv4 VRRP can assist in load balancing outgoing traffic. The figure below shows two virtual routers with their hosts splitting traffic between them. Half of the hosts are configured with a default route to virtual router 1's IP address (10.10.2.250), and the other half are configured with a default route to virtual router 2's IP address (10.10.2.245).

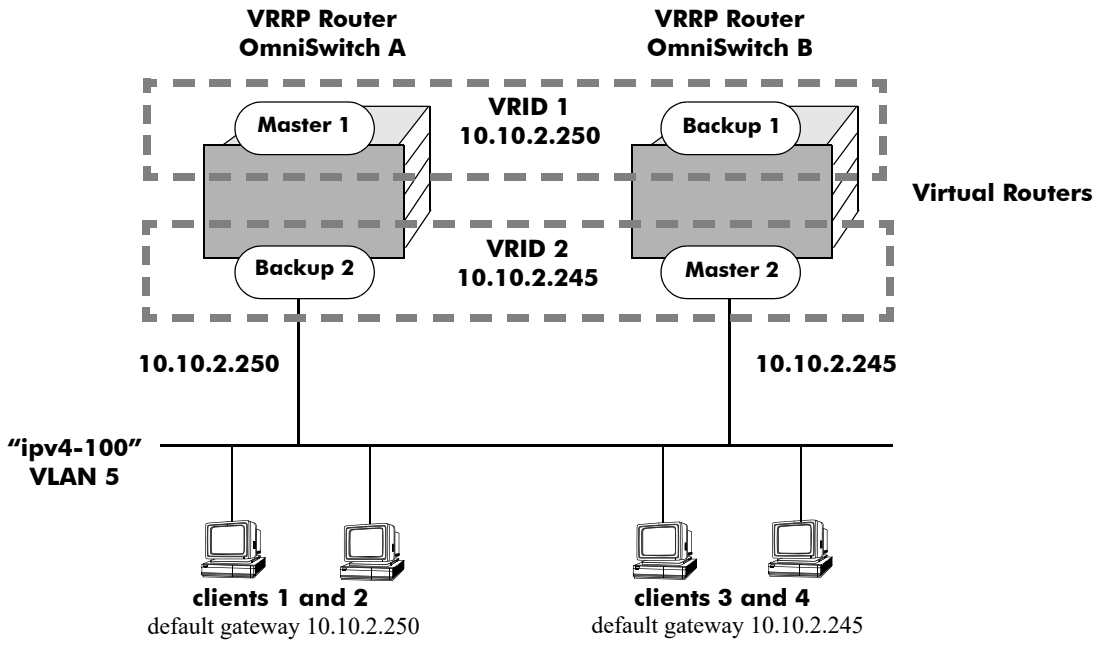


Figure 25-2 : IPv4 VRRP Redundancy and Load Balancing

The CLI commands used to configure this setup are as follows:

1 Create two IPv4 virtual routers for IPv4 interface "ipv4-100", which is bound to VLAN 5. (Note that interface "ipv4-100" must already be created and available on the switch.)

```
-> ip vrrp 1 interface ipv4-100
-> ip vrrp 2 interface ipv4-100
```

2 Configure the IP addresses for each virtual router.

```
-> ip vrrp 1 interface ipv4-100 address 10.10.2.250
-> ip vrrp 2 interface ipv4-100 address 10.10.2.245
```

3 Enable the virtual routers.

```
-> ip vrrp 1 interface ipv4-100 admin-state enable
-> ip vrrp 2 interface ipv4-100 admin-state enable
```

Note. The same IPv4 VRRP configuration must be set up on each switch. The IPv4 VRRP router that contains, or owns, the IP address will automatically become the master for that virtual router. If the IP address is a virtual address, the virtual router with the highest priority will become the master router.

In this scenario, the master of VRID 1 will respond to ARP requests for IP address A using the virtual router MAC address for VRID 1 (00:00:5E:00:01:01). OmniSwitch A is the master for VRID 1 since it contains the physical interface to which 10.10.2.250 is assigned. If OmniSwitch A should become unavailable, OmniSwitch B will become master for VRID 1.

In the same way, the master of VRID 2 will respond to ARP requests for IP address B using the virtual router MAC address for VRID 2 (00:00:5E:00:01:02). OmniSwitch B is the master for VRID 2 since it contains the physical interface to which 10.10.2.245 is assigned. If OmniSwitch B should become unavailable, OmniSwitch A will become master for 10.10.2.245. This configuration provides uninterrupted service for the end hosts.

IPv4 VRRP Tracking Example

The figure below shows two VRRP routers with two virtual routers backing up one IP address on each VRRP router respectively. Virtual router 1 serves as the default gateway on OmniSwitch A for clients 1 and 2 through IP address 10.10.2.250 and virtual router 2 serves as default gateway on OmniSwitch B for clients 3 and 4 through IP address 10.10.2.245. For example, if the port that provides access to the Internet on OmniSwitch A fails, virtual router 1 will continue to be the default router for clients 1 and 2, but clients 1 and 2 will not be able to access the Internet.

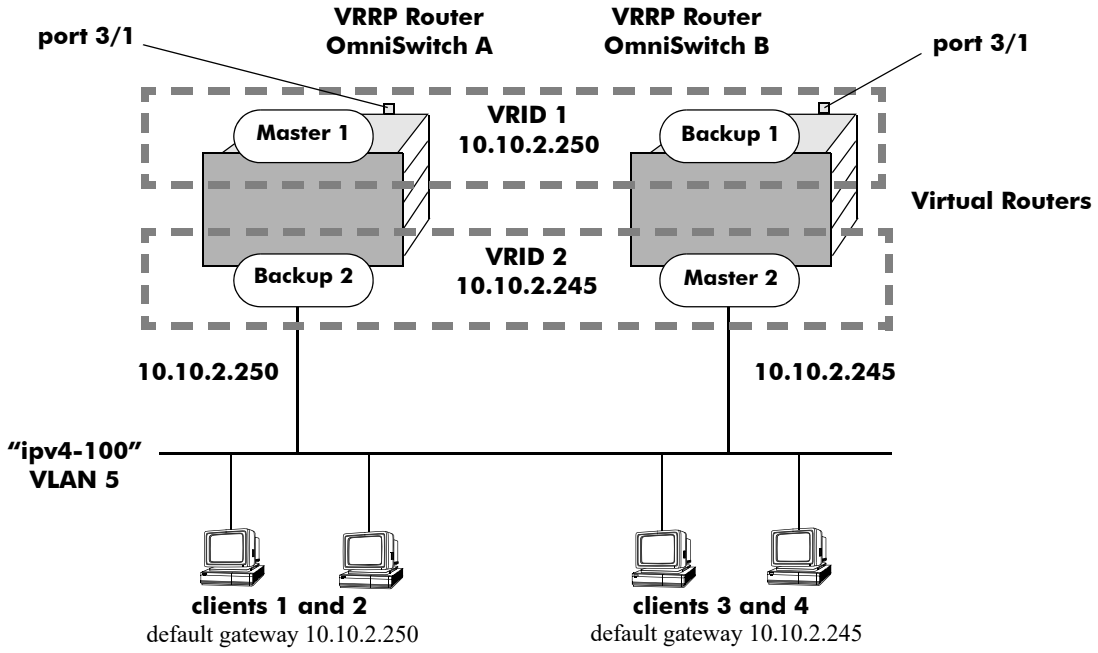


Figure 25-3 : VRRP Tracking Example

In this example, the master for virtual router 1 has a priority of 100 and the backup for virtual router 1 has a priority of 75. The virtual router configuration for VRID 1 and 2 on VRRP router A is as follows:

```
-> ip vrrp 1 interface ipv4-100 priority 100 preempt
-> ip vrrp 2 interface ipv4-100 priority 75
```

The virtual router configuration for VRID 1 and 2 on VRRP router B is as follows:

```
-> ip vrrp 1 interface ipv4-100 priority 75
-> ip vrrp 2 interface ipv4-100 priority 100 preempt
```

To ensure workstation clients 1 and 2 have connectivity to the Internet, configure a tracking policy on VRRP router A to monitor port 3/1 and associate the policy with VRID 1.

```
-> ip vrrp track 1 admin-state enable priority 50 port 3/1
-> ip vrrp 1 interface ipv4-100 track-association 1
```

If port 3/1 on VRRP router A goes down, the master for virtual router A is still functioning but workstation clients 1 and 2 will not be able to get to the Internet. With this tracking policy enabled, however, master router 1's priority will be temporarily decremented to 50, allowing backup router 1 to take over and provide connectivity for those workstations. When port 3/1 on VRRP router A comes backup, master 1 will take over again.

Note. Preempt must be set on switch A virtual router 1, and switch B virtual router 2, in order for the correct master to assume control once their respective ports 3/1 return to viability. In this example, once port 3/1 on switch A is functioning again we want switch A to reestablish itself as the master. See "Setting Preemption for Virtual Routers" on page 25-16 for more information about enabling preemption.

IPv6 VRRP Application Example

In addition to providing redundancy, IPv6 VRRP can assist in load balancing outgoing traffic. The figure below shows two virtual routers with their hosts splitting traffic between them. Half of the hosts are configured with a default route to virtual router 1's IPv6 address (213:100:1::56), and the other half are configured with a default route to virtual router 2's IPv6 address (213:100:1::57).

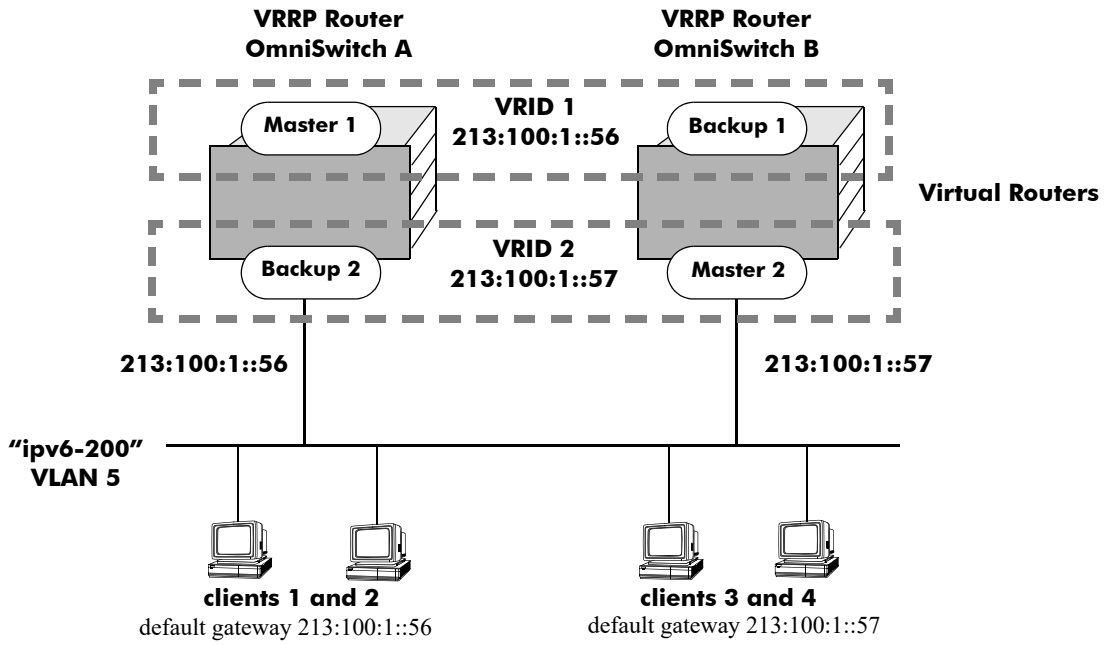


Figure 25-4 : IPv6 VRRP Redundancy and Load Balancing

The CLI commands used to configure this setup are as follows:

1 Create two IPv6 VRRP virtual routers for IPv6 interface "ipv6-200", which is bound to VLAN 5. (Note that interface "ipv6-200" must already be created and available on the switch.)

```
-> ipv6 vrrp 1 interface ipv6-200
-> ipv6 vrrp 2 interface ipv6-200
```

2 Configure the IPv6 addresses for each IPv6 VRRP virtual router.

```
-> ipv6 vrrp 1 interface ipv6-200 address 213:100:1::56
-> ipv6 vrrp 2 interface ipv6-200 address 213:100:1::57
```

3 Enable the IPv6 VRRP virtual routers.

```
-> ipv6 vrrp 1 interface ipv6-200 admin-state enable
-> ipv6 vrrp 2 interface ipv6-200 admin-state enable
```

Note. The same IPv6 VRRP configuration must be set up on each switch. The IPv6 VRRP router that contains, or owns, the IPv6 address will automatically become the master for that virtual router. If the IPv6 address is a virtual address, the virtual router with the highest priority will become the master router.

In this scenario, the master of VRID 1 will respond to neighbor solicitation with a neighbor advertisement for IPv6 address A using the virtual router MAC address for VRID 1 (00:00:5E:00:02:01). OmniSwitch A is the master for VRID 1 since it contains the physical interface to which 213:100:1::56 is assigned. If OmniSwitch A should become unavailable, OmniSwitch B will become master for VRID 1.

In the same way, the master of VRID 2 will respond to neighbor solicitation for IPv6 address B using the virtual router MAC address for VRID 2 (00:00:5E:00:02:02). OmniSwitch B is the master for VRID 2 since it contains the physical interface to which 213:100:1::57 is assigned. If OmniSwitch B should become unavailable, OmniSwitch A will become master for 213:100:1::57. This configuration provides uninterrupted service for the end hosts.

IPv6 VRRP Tracking Example

The figure below shows two IPv6 VRRP routers with two virtual routers backing up one IPv6 address on each VRRP router respectively. Virtual router 1 serves as the default gateway on OmniSwitch A for clients 1 and 2 through IPv6 address 213:100:1::56. For example, if the port that provides access to the Internet on OmniSwitch A fails, virtual router 1 will continue to be the default router for clients 1 and 2, but clients 1 and 2 will not be able to access the Internet.

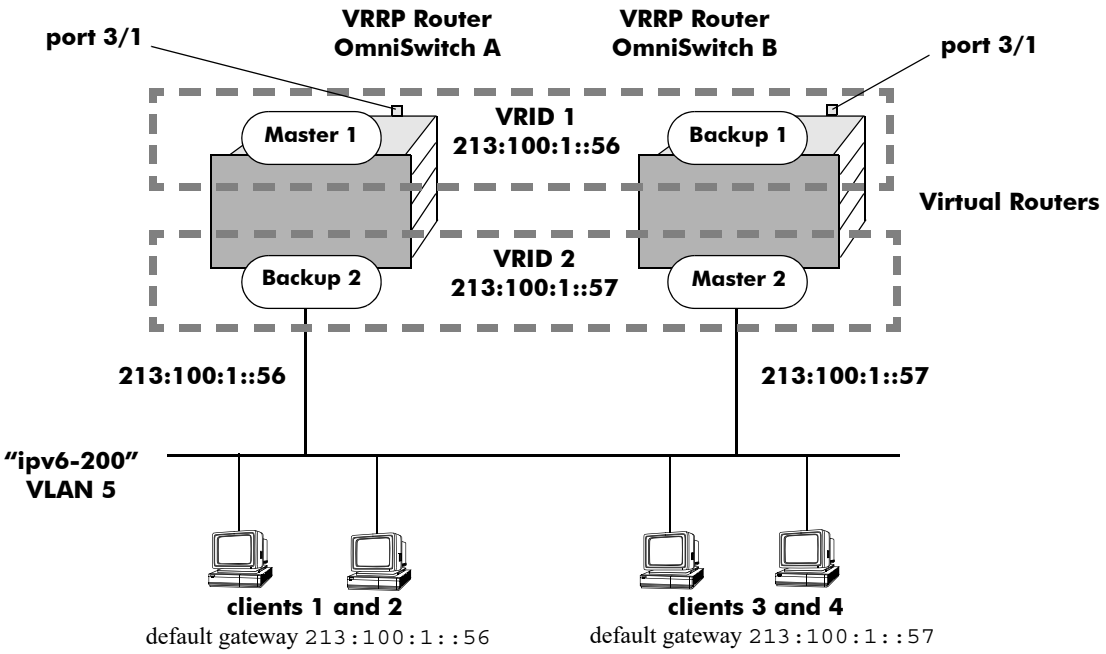


Figure 25-5 : VRRP Tracking Example

In this example, the master for virtual router 1 has a priority of 100 and the backup for virtual router 1 has a priority of 75. The virtual router configuration for VRID 1 and 2 on VRRP router A is as follows:

```
-> ipv6 vrrp 1 interface ipv6-200 priority 100 preempt
-> ipv6 vrrp 2 interface ipv6-200 priority 75
```

The virtual router configuration for VRID 1 and 2 on VRRP router B is as follows:

```
-> ipv6 vrrp 1 interface ipv6-200 priority 75
-> ipv6 vrrp 2 interface ipv6-200 priority 100 preempt
```

To ensure workstation clients 1 and 2 have connectivity to the Internet, configure a tracking policy on VRRP router A to monitor port 3/1 and associate the policy with VRID 1.

```
-> ip vrrp track 1 admin-state enable priority 50 port 3/1
-> ipv6 vrrp 1 interface ipv6-200 track-association 1
```

If port 3/1 on VRRP router A goes down, the master for virtual router A is still functioning, but workstation clients 1 and 2 will not be able to get to the Internet. With this tracking policy enabled, however, master router 1's priority will be temporarily decremented to 50, allowing backup router 1 to take over and provide connectivity for those workstations. When port 3/1 on VRRPv3 router A comes backup, master 1 will take over again.

Note. Preempt must be set on switch A virtual router 1, and switch B virtual router 2, in order for the correct master to assume control once their respective ports 3/1 return to viability. In this example, once port 3/1 on switch A is functioning again we want switch A to reestablish itself as the master. See "Setting Preemption for Virtual Routers" on page 25-16 for more information about enabling preemption.

---

26 Configuring Server Load Balancing

The OmniSwitch implementation of Server Load Balancing (SLB) software provides a method to logically manage a group of physical servers sharing the same content (known as a *server farm*) as one large virtual server (known as an *SLB cluster*). SLB clusters are identified and accessed using either a Virtual IP (VIP) address or a QoS policy condition. Traffic is always routed to VIP clusters and either bridged or routed to policy condition clusters. The OmniSwitch operates at wire speed to process client requests and then forward them to the physical servers within the cluster.

Using SLB clusters can provide cost savings (costly hardware upgrades can be delayed or avoided), scalability (as the demands on your server farm grow you can add additional physical servers), reliability (if one physical server goes down the remaining servers can handle the remaining workload), and flexibility (you can tailor workload requirements individually to servers within a cluster).

In This Chapter

This chapter describes the basic components of Server Load Balancing and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Procedures to configure SLB on a switch on page 26-10.
- Procedures to configure logical SLB clusters on page 26-11.
- Procedures to configure physical servers in SLB clusters on page 26-13.
- Procedures to configure SLB probes on page 26-17.
- Procedures for troubleshooting and maintenance on page 26-16 and page 26-21.

Server Load Balancing Default Values

The following table lists default values for the SLB software:.

Parameter Description	Command	Default Value/Comments
Global SLB administrative status	**ip slb admin-state**	Disabled
Ping period	**ip slb cluster ping period**	60 seconds
Ping timeout	**ip slb cluster ping timeout**	3000 milliseconds
Ping retries	**ip slb cluster ping retries**	3
Administrative status of an SLB cluster	**ip slb cluster admin-state**	Enabled
Administrative status of physical servers in an SLB cluster	**ip slb server ip cluster**	Enabled
Relative weight of a physical server in an SLB cluster	**ip slb server ip cluster**	1
SLB probes configured	**ip slb probe**	None configured
SLB probe timeout	**ip slb probe timeout**	3000 seconds
SLB probe period	**ip slb probe period**	60 seconds
SLB probe port number	**ip slb probe port**	0
SLB probe retries	**ip slb probe retries**	3
SLB probe user name	**ip slb probe username**	None configured
SLB probe password	**ip slb probe password**	None configured
SLB probe URL	**ip slb probe url**	None configured
SLB probe expected status	**ip slb probe status**	200
SLB probe send string	**ip slb probe send**	None configured
SLB probe expect string	**ip slb probe expect**	None configured

Quick Steps for Configuring Server Load Balancing

Follow the steps below for a quick tutorial on configuring parameters for SLB. Additional information on how to configure each command is given in the subsections that follow. Note that this example configures a VIP cluster. See the tutorial on page 26-4 for quick steps on configuring a QoS policy condition cluster.

1 Enable SLB globally with the **ip slb admin-state** command as shown below:

```
-> ip slb admin-state enable
```

2 Configure the SLB VIP cluster using the **ip slb cluster** command with the **vip** parameter. For example:

```
-> ip slb cluster WorldWideWeb vip 128.241.130.204
```

3 Assign physical servers to the SLB cluster and specify a relative weight for each server (default value for weight is 1) with the **ip slb server ip cluster** command. For example:

```
-> ip slb server ip 128.241.130.127 cluster WorldWideWeb
-> ip slb server ip 128.241.130.109 cluster WorldWideWeb weight 4
-> ip slb server ip 128.241.130.115 cluster WorldWideWeb weight 6
-> ip slb server ip 128.241.130.135 cluster WorldWideWeb admin-state disable
weight 8
```

As an option, you can verify your SLB settings by entering **show ip slb cluster** followed by the name of the SLB cluster. For example:

```
-> show ip slb cluster WorldWideWeb

Cluster WorldWideWeb
  VIP                              : 128.241.130.204,
  Type                             : L3,
  Admin status                     : Enabled,
  Operational status               : In Service,
  Ping period (seconds)            : 60,
  Ping timeout (milliseconds)      : 3000,
  Ping retries                     : 3,
  Probe                            : None,
  Number of packets                : 3800,
  Number of servers                : 4
  Server 128.241.130.109
     Admin status = Enabled, Operational Status = In Service,
     Weight = 4, Availability (%) = 100
  Server 128.241.130.115
     Admin status = Enabled, Operational Status = In Service,
     Weight = 6, Availability (%) = 98
  Server 128.241.130.127
     Admin status = Enabled, Operational Status = Discovery,
     Weight = 1, Availability (%) = 0
  Server 128.241.130.135
     Admin status = Disabled, Operational Status = Disabled,
     Weight = 8, Availability (%) = 0
```

An example of what these configuration commands look like entered sequentially on the command line:

```
-> ip slb admin-state enable
-> ip slb cluster WorldWideWeb vip 128.241.130.204
-> ip slb server ip 128.241.130.127 cluster WorldWideWeb
-> ip slb server ip 128.241.130.109 cluster WorldWideWeb weight 4
-> ip slb server ip 128.241.130.115 cluster WorldWideWeb weight 6
-> ip slb server ip 128.241.130.135 cluster WorldWideWeb admin-state disable
weight 8
```

Quick Steps for Configuring a QoS Policy Condition Cluster

Follow the steps below for a quick tutorial on how to configure a QoS policy condition cluster:

1 Create the QoS policy condition that classifies traffic for the SLB cluster. For example:

```
-> policy network group SOURCE 100.0.0.1 100.0.0.2 100.0.0.3 100.0.0.4
-> policy condition c1 source network group SOURCE destination tcp-port 80
-> qos apply
```

2 Configure the SLB cluster using the **ip slb cluster** command with the **condition** parameter. For example:

```
-> ip slb cluster Intranet condition c1
```

3 Assign physical servers to the SLB condition cluster and specify a relative weight for each server (default value for weight is 1) with the **ip slb server ip cluster** command. For example:

```
-> ip slb server ip 103.10.50.1 cluster Intranet
-> ip slb server ip 103.10.50.2 cluster Intranet weight 4
-> ip slb server ip 103.10.50.3 cluster Intranet admin-state disable weight 2
```

Note. As an option, you can configure an SLB server as a backup server. See "Configuring a Server in an SLB Cluster as a Backup Server" on page 26-15 for more information.

As an option, you can verify your SLB settings by entering **show ip slb cluster** followed by the name of the SLB cluster. For example:

```
-> show ip slb cluster Intranet

Cluster Intranet
VIP                             : 123.12.1.2,
Type                            : L3
Admin status                    : Enabled,
  Operational status            : In Service,
  Ping period (seconds)         = 60,
  Ping timeout (milliseconds)   = 3000,
  Ping retries                  = 3,
  Probe                         = None,
  Number of packets             = 10000,
  Number of servers             = 2
    Server 103.10.50.1
      Admin status = Enabled, Operational status = In Service,
      Weight = 1, Availability (%) = 100
    Server 103.10.50.2
      Admin status = Enabled, Operational status = In Service,
      Weight = 4, Availability (%) =  99
```

```
    Server 103.10.50.3
      Admin status = Disabled, Operational status = Disabled,
      Weight = 2, Availability (%) = 0
```

As an option, you can also display traffic statistics for an SLB condition cluster by entering **show ip slb cluster** followed by the cluster name and the **statistics** parameter. For example, the following command displays the packet count for traffic that is classified for the "Intranet" cluster:

```
-> show ip slb cluster Intranet statistics

                          Admin    Operational
Cluster Name              Status   Status                Count
-----------------------+--------+--------------------+--------
Intranet                 Enabled  In Service           2 Servers
      Src IP 100.0.0.1/255.255.255.255                      2500
      IP Dst TCP Port 80
      Src IP 100.0.0.2/255.255.255.255                      2500
      IP Dst TCP Port 80
      Src IP 100.0.0.3/255.255.255.255                      2500
      IP Dst TCP Port 80
      Src IP 100.0.0.4/255.255.255.255                      2500
      IP Dst TCP Port 80
```

An example of what the configuration commands look like entered sequentially on the command line:

```
-> policy network group SOURCE 100.0.0.1 100.0.0.2 100.0.0.3 100.0.0.4
-> policy condition c1 source network group SOURCE destination tcp-port 80
-> qos apply
-> ip slb cluster Intranet condition cl
-> ip slb server ip 103.10.50.1 cluster Intranet
-> ip slb server ip 103.10.50.2 cluster Intranet weight 4
-> ip slb server ip 103.10.50.3 cluster Intranet admin-state disable weight 2
```

You can verify your SLB settings by entering **show ip slb cluster server** followed by the name of the SLB cluster. For example:

```
-> show ip slb cluster Intranet server 103.10.50.3

Cluster Intranet
  VIP 103.10.50.50
    Server 103.10.50.3
      Admin status                            : Disabled,
      Oper status                             : In Service,
      Probe                                   = None,
      Admin weight                            = 2,
      Availability time (%)                   = 98,
      Ping failures                           = 0,
      Last ping round trip time (milliseconds) = 1,
      Probe status                            = OK,
```

Note. Once a cluster is created, the Virtual IP or condition cannot be modified. To modify these values, delete the cluster and re-create the cluster with the different VIP and conditions.

Server Load Balancing Overview

The following sections describe SLB operational theory (see "Server Load Balancing Cluster Identification" on page 26-6), an SLB example ("Server Load Balancing Example" on page 26-7), and server health monitoring (see "Server Health Monitoring" on page 26-9).

Note. The OmniSwitch also offers link aggregation, which combines multiple Ethernet links into one virtual channel. Please refer to Chapter 10, "Configuring Dynamic Link Aggregation,"for more information on link aggregation and dynamic link aggregation, and to Chapter 9, "Configuring Static Link Aggregation," for information on static (OmniChannel) link aggregation.

Server Load Balancing Cluster Identification

An SLB cluster consists of a group of physical servers, also known as a server farm. The SLB cluster appears as one large virtual server, which is identified using one of the following methods:

- Virtual IP (VIP)—An IP address is assigned to the cluster (virtual server). Client requests destined for this VIP are routed (Layer-3 mode) to the servers that are members of the VIP cluster. Note that it is necessary to configure cluster servers with a loopback interface.
- Condition—A QoS policy condition name is assigned to the cluster (virtual server). Client requests that meet the criteria of the policy condition are bridged (Layer-2 mode) or routed (Layer-3 mode) to the servers that are members of the condition cluster. Note that it is *not* necessary to configure cluster servers with a loopback interface.

Note. See "Configuring and Deleting SLB Clusters" on page 26-11 for more information on configuring VIP and condition clusters.

Server Load Balancing Cluster Modes

The cluster mode refers to whether client requests are bridged (Layer-2 mode) or routed (Layer-3 mode) by the switch to the appropriate SLB cluster. A VIP cluster only supports Layer-3 mode, so request packets are always routed to the cluster. A condition cluster supports both Layer-2 *and* Layer-3 modes.

When the Layer-3 mode is active (VIP or condition clusters), routed packets are modified as follows:

- The source IP address is changed to the IP address for the switch router interface.
- The destination IP address is changed to the IP address of the destined server.
- The TTL value is decremented.

When the Layer-2 mode is active (condition clusters only), request packets are not modified and are only switched within the same VLAN domain. The Layer-2 or Layer-3 mode is selected when the condition cluster is configured on the switch. See "Configuring an SLB Cluster with a QoS Policy Condition" on page 26-11 for more information.

Server Load Balancing Example

In the figure on the following page, an SLB cluster consisting of four (4) physical servers has been configured with a VIP of 128.241.130.204 and an SLB cluster name of “WorldWideWeb.” The switch processes requests sent by clients to the VIP of 128.241.130.204 and sends to the appropriate physical server, depending on configuration and the operational states of the physical servers. The switch then transmits the requested data from the physical server back to the client.

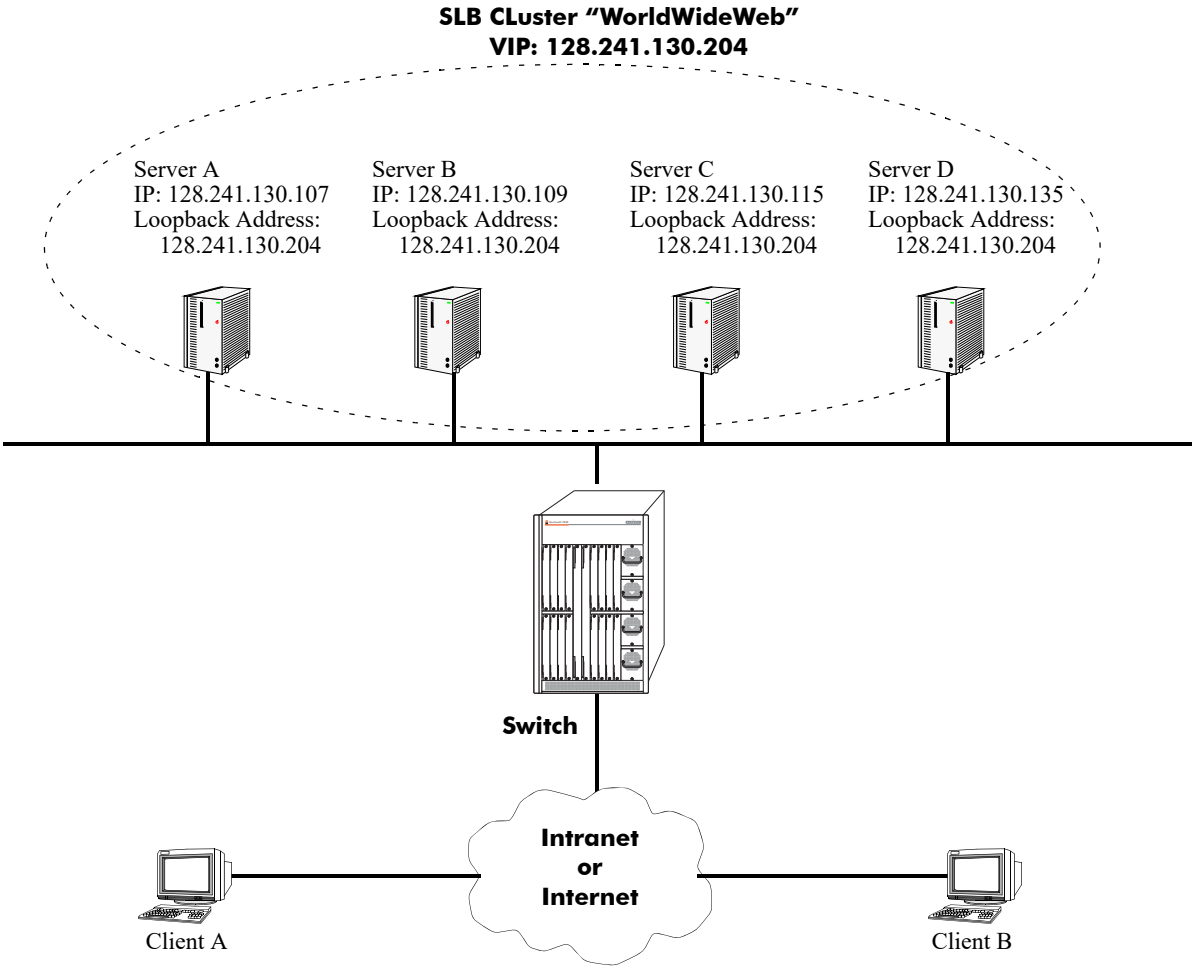


Figure 26-1 : Example of a Server Load Balancing (SLB) Cluster

Weighted Round Robin Distribution Algorithm

In order to distribute traffic among operating servers, the SLB cluster must have a method of selecting a server among a pool (cluster) of operating servers ("in service" mode) depending on some criteria. This method is commonly called the SLB cluster *distribution algorithm.*

The distribution algorithm on an OmniSwitch is weighted round robin, where the SLB cluster distributes traffic according to the relative "weight" a server has within an SLB cluster. In the figure below, for example, Server A has been assigned by the network administrator a relative weight of 30, which is the largest weight in the SLB cluster called "WorldWideWeb." Server A handles twice as much traffic as Server C (which has a weight of 15), three times the traffic of Server D (which has a weight of 10), and six times the traffic of Server B (which has a weight of 5).

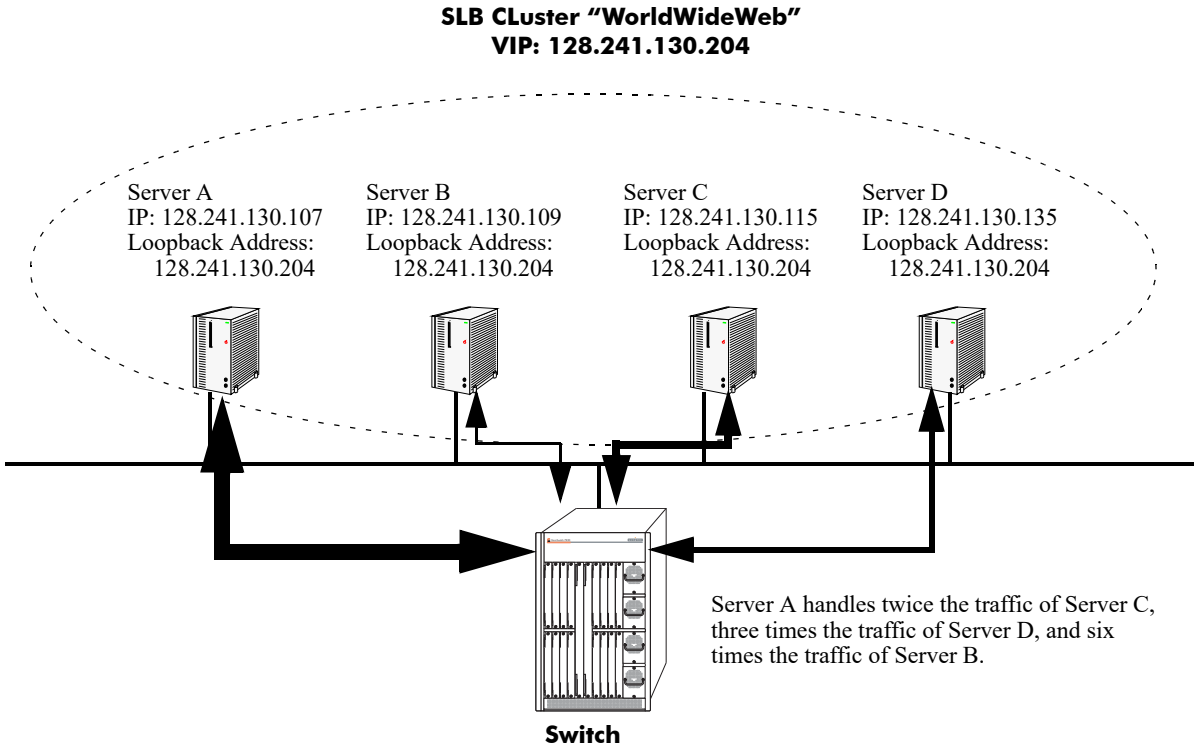


Figure 26-2 : Weighted Round Robin Algorithm

Note. See "Modifying the Relative Weight of a Physical Server" on page 26-15 for information on modifying the relative weights of servers in an SLB cluster.

Server Health Monitoring

The OmniSwitch Server Load Balancing (SLB) software performs checks on the links from the switch to the servers. In addition, the SLB software also sends ICMP echo requests (ping packets) to the physical servers to determine their availability.

Note. You can use the **show ip slb cluster server** command, which is described in "Displaying Server Load Balancing Status and Statistics" on page 26-21, to display link and ping status of physical servers.

These health checks performed by the switch are used by the SLB software to determine the operational states of servers. The possible operational states are described in the table below:

Operational States

Disabled	The server has been administratively disabled by the user.
No Answer	The server has not responded to ping requests from the switch.
Link Down	There is a bad connection to the server.
Discovery	The switch is pinging a physical server.
In Service	The server can be used for client connections.
Retrying	The switch is making another attempt to bring up the server.

In Release 5.1.6 and later you can configure probes to monitor the health of clusters and servers. See "Configuring SLB Probes" on page 26-17 for more information.

Configuring Server Load Balancing on a Switch

This section describes how to use the OmniSwitch Command Line Interface (CLI) commands to configure Server Load Balancing (SLB) on a switch.

Note. See "Quick Steps for Configuring Server Load Balancing" on page 26-3 for a brief tutorial on configuring these mandatory parameters.

When configuring SLB parameters for an SLB cluster, you must perform the following steps:

1 Enable Server Load Balancing on Your Switch. To enable Server Load Balancing (SLB) on a switch, use the **ip slb admin-state** command, which is described in "Enabling and Disabling Server Load Balancing" on page 26-10.

2 Configure the Logical Server Load Balancing Cluster. To configure a logical SLB cluster, use the **ip slb cluster** command, which is described in "Configuring and Deleting SLB Clusters" on page 26-11.

3 Assign Physical Servers to the Logical Server Load Balancing Cluster. To add physical servers to a logical SLB cluster, use the **ip slb server ip cluster** command, which is described in "Assigning Servers to and Removing Servers from a Cluster" on page 26-13.

Note. Routing (which is enabled by default) must be enabled on a switch or Server Load Balancing will not operate.

The OmniSwitch implementation of SLB software is preconfigured with the default values shown in the table in "Server Load Balancing Default Values" on page 26-2. Depending on the requirements of your network and server farm, you may need to configure more parameters than the mandatory ones described in this section. See "Modifying Optional Parameters" on page 26-14 for information on configuring additional SLB parameters.

Enabling and Disabling Server Load Balancing

By default, Server Load Balancing (SLB) is disabled on a switch. The following subsections describe how to enable and disable SLB on a switch with the **ip slb admin-state** command.

Note. You must enable or disable Server Load Balancing on an entire switch. You cannot enable SLB on a per port or per slot basis.

Enabling SLB

To enable SLB switch wide, use the **ip slb admin-state** command by entering:

```
-> ip slb admin-state enable
```

Disabling SLB

To disable SLB switch wide, use the **ip slb admin-state** command by entering:

```
-> ip slb admin-state disable
```

Configuring and Deleting SLB Clusters

The following subsections describe how to configure and delete SLB clusters with the **ip slb cluster** command.

Configuring an SLB Cluster with a VIP Address

Consider the following when configuring a VIP cluster:

- Specify a cluster name that is at least 1 character and less than or equal to 23 characters long.
- To use spaces in an SLB cluster name, enclose the entire name within quotation marks (for example, “web server”).
- The VIP address of the SLB cluster *must* be an address in the same subnet as the servers.
- VIP only supports the Layer-3 SLB mode, which is enabled by default.

To configure an SLB cluster that uses VIP classification to bridge or route client requests to the cluster servers, use the **ip slb cluster** command with the **vip** parameter. For example, to configure an SLB cluster called “Web_Server” with a VIP address of 10.123.11.14, you would enter:

```
-> ip slb cluster Web_Server vip 10.123.11.14
```

Configuring an SLB Cluster with a QoS Policy Condition

Consider the following when configuring a QoS policy condition cluster:

- Specify a cluster name that is at least 1 character and less than or equal to 23 characters long.
- To use spaces in an SLB cluster name, enclose the entire name within quotation marks (for example, “web server2”).
- The QoS policy condition name specified must the switch configuration.

To configure an SLB cluster that uses a QoS policy condition to qualify client requests for bridging or routing to the cluster servers, use the **ip slb cluster** command with the **condition** parameter and either the **l2** or **l3** parameter. For example, to configure an SLB cluster called “Web_Server2” with the “cond1” policy condition and using the L2 mode, you would enter:

```
-> ip slb cluster Web_Server2 condition cond1 l2
```

How to Create a QoS Policy Condition

Use the **policy condition** command to create a QoS policy condition. For example, the following command creates a source port condition named “cond1”:

```
-> policy condition cond1 source port 1/24
```

The condition created in the above example, “cond1”, uses the source port value to classify traffic. When this same condition is associated with an SLB cluster, client requests received on the specified source port are then sent to a server that is a member of the associated cluster.

The following QoS policy conditions are supported individually and in combination with each other when used to configure SLB condition clusters:

QoS Policy Condition Keywords		
source vlan	**tos**	**ethertype**
source port	**dscp**	**protocol**
destination port	**802.1p**	**source tcp-port**
source port group	**source ip address**	**destination tcp-port**
destination port group	**destination ip address**	**source udp-port**
source mac	**source network group**	**destination udp-port**
destination mac	**destination network group**	**icmp type**
source mac group	**service**	**icmp code**
destination mac group	**service group**	**tcp flags**

See Chapter 28, "Configuring QoS," for more information about configuring and displaying QoS policy conditions.

Automatic Configuration of SLB Policy Rules

When you configure an SLB cluster, a Quality of Service (QoS) policy condition, action, and rule are automatically configured for it. In addition, the switch software automatically names the condition, action, and rule by adding the prefix **SLB-cond-**, **SLB-act-**, and **SLB-rule-**, respectively, to the name of the SLB cluster for each name.

For example, if you configured an SLB cluster called "Web_Server" a policy condition called "SLB-cond-Web_Server," a policy action called "SLB-act-Web_Server," and a policy rule called "SLB-rule-Web_Server" would be created.

Note that the user-configured policy condition associated with an SLB cluster is the condition used for the automatically configured SLB policy rule. For example, if you configured an SLB cluster called "Web_Server2" and associated it with the "cond1" condition, a policy rule called "SLB-rul-Web-Server2" would be created with the "cond1" condition and the "SLB-act-Web_Server2" action.

You can display QoS policy rules with the **show policy rule** command. To use this command, enter **show policy rule** followed by the name of the rule. For example, the following commands display the policy rule called "SLB-rul-Web_Server" and the policy rule called "SLB-rul-Web_Server2":

```
-> show policy rule SLB-rul-Web-Server
Rule name                        = SLB-rul-Web-Server,
  From                           = api,
  Precedence                     = 65000,
  Condition name                 = SLB-cnd-Web-Server,
  Action name                    = SLB-act-Web-Server
```

You can also use the **show policy condition** command to display policy conditions and the **show policy action** command to display policy actions. See Chapter 28, "Configuring QoS," for more information on configuring and displaying QoS policies.

Deleting an SLB Cluster

To delete an SLB cluster, use the **no** form of the **ip slb reset statistics** command by entering **no ip slb cluster** followed by the name of the cluster.

For example, to delete an SLB called "Web_Server", you would enter:

```
-> no ip slb cluster Web_Server
```

Note. When you delete an SLB cluster you also delete the QoS policy, condition, and rule associated with the cluster.

Assigning Servers to and Removing Servers from a Cluster

The following subsections describe how to assign servers to an SLB cluster and how to remove servers from an SLB cluster with the **ip slb server ip cluster** command.

Note. You can also use the **ip slb server ip cluster** command to administratively disable or enable a server (see "Taking a Server On/Off Line" on page 26-16).

Assigning a Server to an SLB Cluster

You assign physical servers to an existing logical SLB cluster with the **ip slb server ip cluster** command by entering **ip slb server ip**, the IP address of the server in dotted decimal format, **cluster**, and the name of the SLB cluster.

For example, to assign a server with an IP address of 10.105.16.118 to an SLB cluster called "Web_Server", you would enter:

```
-> ip slb server ip 10.105.16.118 cluster Web_Server
```

For example, to assign three physical servers with IP addresses of 10.105.16.121, 10.105.16.122, and 10.105.16.123, respectively, to an SLB cluster called "Web_Server", enter the following CLI commands:

```
-> ip slb server ip 10.105.16.121 cluster Web_Server
-> ip slb server ip 10.105.16.122 cluster Web_Server
-> ip slb server ip 10.105.16.123 cluster Web_Server
```

Removing a Server from an SLB Cluster

To remove a physical server from an SLB cluster, use the **no** form of the **ip slb server ip cluster** command by entering **no ip slb server ip**, the IP address of the server you want to remove in dotted decimal format, **cluster**, and the name of the SLB cluster.

For example, to remove a server with an IP address of 10.105.16.121 from an SLB cluster called "Web_Server" you would enter:

```
-> no ip slb server ip 10.105.16.121 cluster Web_Server
```

Modifying Optional Parameters

As shown in the table on page 26-2, The OmniSwitch implementation of SLB software is preconfigured with default values for the SLB cluster's "sticky" time, ping timeout, ping period, ping retries, and relative weight (preference). The following subsections describe how to modify these parameters.

- **Modifying the Ping Period**. You can modify the ping period with the **ip slb cluster ping period** command, which is described in "Modifying the Ping Period" on page 26-14.
- **Modifying the Ping Timeout**. You can modify the ping timeout with the **ip slb cluster ping timeout** command, which is described in "Modifying the Ping Timeout" on page 26-14.
- **Modifying the Number of Ping Retries**. You can modify the number of ping retries with the **ip slb cluster ping retries** command, which is described in "Modifying the Ping Retries" on page 26-15.
- **Modifying the Relative Weight of an SLB Cluster Server**. You can configure server preferences within a cluster by modifying the relative weight of each server with the **ip slb server ip cluster** command, which is described in "Modifying the Relative Weight of a Physical Server" on page 26-15.

Modifying the Ping Period

You can modify this value with the **ip slb cluster ping period** command by entering **ip slb cluster**, the name of the SLB cluster, **ping period**, and the user-specified number of seconds. For default and range of values for the parameters, check the "Server Load Balancing Default Values" on page 26-2 and the "Server Load Balancing Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

For example, to set the ping period on an SLB cluster called "Web_Server" to 1200 seconds enter:

```
-> ip slb cluster Web_Server ping period 120
```

Note. If you set the ping period to any value other than 0, then the ping period must be greater than or equal to the ping timeout value divided by 1000. For example, if the ping timeout is 5000 milliseconds, the ping period must be at least 5 seconds. The ping timeout value can be modified with the **ip slb cluster ping timeout** command, which is described in "Modifying the Ping Timeout" on page 26-14.

Modifying the Ping Timeout

You can modify the value of the ping period with the **ip slb cluster ping timeout** command by entering **ip slb cluster**, the name of the SLB cluster, **ping timeout**, and the user-specified number of milliseconds.

For example to set the ping timeout on an SLB cluster called "Web_Server" to 1000 milliseconds enter:

```
-> ip slb cluster Web_Server ping timeout 1000
```

Note. You can modify the ping period with the **ip slb cluster ping period** command, which is described in "Modifying the Ping Period" on page 26-14.

Modifying the Ping Retries

You can modify the ping retry value with the **ip slb cluster ping retries** command by entering **ip slb cluster**, the name of the SLB cluster, **ping retries**, and the user-specified number of ping retries. For example:

```
-> ip slb cluster Web_Server ping retries 5
```

Modifying the Relative Weight of a Physical Server

To modify the relative weight of a server, use the **ip slb server ip cluster** command by entering **ip slb server**, the IP address of the physical server you want to modify, **cluster**, the name of the SLB cluster to which this server belongs, **weight**, and the value for the relative weight, (the switch prevents the physical server from being assigned any new connections), with 32 having the greatest relative weight.

For example, to set the relative weight of a server with an IP address of 10.105.16.121 that belongs to an SLB cluster called "Web_Server" to 5 enter:

```
-> ip slb server ip 10.105.16.121 cluster Web_Server weight 5
```

Server weights are relative. For example, if Servers A and B have respective weights of 5 and 10 within a cluster, Server A would get half the traffic of server B. Since weights are relative, assigning Servers A and B respective weights of 1 and 2, or 5 and 10, etc., would produce identical results.

Note. The **ip slb server ip cluster** command is also used to add or remove servers from an SLB cluster (see "Assigning Servers to and Removing Servers from a Cluster" on page 26-13) and for administratively enabling and disabling a server in an SLB cluster (see "Taking a Server On/Off Line" on page 26-16).

Configuring a Server in an SLB Cluster as a Backup Server

You can configure a server in a cluster as a backup server with the **ip slb server ip cluster weight** command by entering **ip slb server ip**, the IP address of the server, **cluster**, the name of the SLB cluster, **weight** and weight value as zero.

For example, to configure a server with an IP address of 10.105.16.118 in an SLB cluster called "Web_Server" as a backup server, enter:

```
-> ip slb server ip 10.105.16.118 cluster Web_Server weight 0
```

Assigning a weight of 0 (zero) to a server prevents this server from being assigned any new connections.This server becomes a backup server.

Taking Clusters and Servers On/Off Line

Server Load Balancing (SLB) **show** commands provide tools to monitor traffic and troubleshoot problems. These commands are described in "Displaying Server Load Balancing Status and Statistics" on page 26-21. If problems are identified, you can use the **ip slb cluster admin-state** command to administratively disable an entire SLB cluster or the **ip slb server ip cluster** command to administratively disable individual servers within an SLB cluster. These commands are described in the following sections.

Taking a Cluster On/Off Line

The following subsections describe how to bring an SLB cluster on line and how to take it off line with the **ip slb cluster admin-state** command.

Bringing an SLB Cluster On Line

You can bring an administratively disabled SLB cluster on line with the **ip slb cluster admin-state** command by entering **ip slb cluster**, the name of the SLB cluster, and **admin-state enable**.

For example, to bring an SLB cluster called "WorldWideWeb" on line, you would enter:

```
-> ip slb cluster WorldWideWeb admin-state enable
```

Taking an SLB Cluster Off Line

You can take a Server Load Balancing (SLB) cluster off line with the **ip slb cluster admin-state** command by entering **ip slb cluster**, the name of the SLB cluster, and **admin-state disable**.

For example, to take an SLB cluster called "WorldWideWeb" off line, you would enter:

```
-> ip slb cluster WorldWideWeb admin-state disable
```

Taking a Server On/Off Line

The following subsections describe how to bring a physical server on line and how to take it off line with the **ip slb server ip cluster** command.

Note. The **ip slb server ip cluster** command is also used to add or remove physical servers from an SLB cluster (see "Assigning Servers to and Removing Servers from a Cluster" on page 26-13).

Bringing a Server On Line

You bring an administratively disabled server in an SLB cluster on line with the **ip slb server ip cluster** command by entering **ip slb server**, the IP address of the server you want to enable in dotted decimal format, **cluster**, the name of the SLB cluster to which the server belongs, and **admin-state enable**.

For example, to administratively enable a server with an IP address of 10.105.16.121 that belongs to an SLB cluster called "Web_Server", you would enter:

```
-> ip slb server ip 10.105.16.121 cluster Web_Server admin-state enable
```

Taking a Server Off Line

You can administratively disable a server in an SLB cluster and take it off line with the **ip slb server ip cluster** command by entering **ip slb server**, the IP address of the server you want to disable in dotted decimal format, **cluster**, the name of the SLB cluster to which the server belongs, and **admin-state disable**.

For example, to administratively disable a server with an IP address of 10.105.16.123 that belongs to an SLB cluster called "Web_Server", you would enter:

```
-> ip slb server ip 10.105.16.123 cluster Web_Server admin-state disable
```

Configuring SLB Probes

Server Load Balancing (SLB) probes allow you to check the health of logical clusters and physical servers. Supported features include:

- Support for server health monitoring using Ethernet link state detection
- Support for server health monitoring using IPv4 ICMP ping
- Support for server health monitoring using a Content Verification Probe

Creating SLB Probes

To create an SLB probe use the **ip slb probe** command by entering the command followed by the user-configured probe name and the probe type, which can be any one of the following listed in the table below:

ip slb probe keywords		
ftp	**http**	**https**
imap	**imaps**	**nntp**
ping	**pop**	**pops**
smtp	**tcp**	**udp**

For example, to create an HTTP SLB probe called "server_probe1", enter:

```
-> ip slb probe server_probe1 http
```

You can configure up to 20 probes on a switch.

Deleting SLB Probes

To delete an SLB use the **no** form of the **ip slb probe** command by entering **no ip slb probe** followed by the probe name. For example, to delete an SLB probe called "server_probe1", enter:

```
-> no ip slb probe server_probe1
```

Associating a Probe with a Cluster

To associate an existing SLB probe with a cluster use the **ip slb cluster probe** command by entering **ip slb cluster** followed by the user-configured cluster name, **probe**, and the user-configured probe name.

For example, to associate a probe called "cluster_probe1" with a cluster called "WorldWideWeb", enter:

```
-> ip slb cluster WorldWideWeb probe cluster_probe1
```

Associating a Probe with a Server

To associate an existing SLB probe with a server use the **ip slb server ip cluster probe** command by entering **ip slb server ip** followed by IP address of the server, **cluster**, the user-configured cluster name, **probe**, and the user-configured probe name.

For example, to associate a probe called "server_probe1" with a server with an IP address of 10.255.11.127 that belongs to a cluster called "WorldWideWeb", enter:

```
-> ip slb server ip 10.255.11.127 cluster WorldWideWeb probe server_probe1
```

Modifying SLB Probes

The following subsections describe how to modify existing SLB probes.

Modifying the Probe Timeout

To modify this value, use the **ip slb probe timeout** command by entering **ip slb probe** followed by the user-configured probe name, the probe type, **timeout**, and the user-specified timeout value.

Note. See "Creating SLB Probes" on page 26-17 for a list of valid probe types.

For example, to set the timeout for an HTTP SLB probe called "server_probe1" to 12000 seconds, enter:

```
-> ip slb probe server_probe1 http timeout 12000
```

Modifying the Probe Period

To modify this value, use the **ip slb probe period** command by entering **ip slb probe** followed by the user-configured probe name, the probe type, **period**, and the user-specified period value.

Note. See "Creating SLB Probes" on page 26-17 for a list of valid probe types.

For example, to set the period for an HTTP SLB probe called "server_probe1" to 120 seconds, enter:

```
-> ip slb probe server_probe1 http period 120
```

Modifying the Probe TCP/UDP Port

To modify this value, use the **ip slb probe port** command by entering **ip slb probe** followed by the user-configured probe name, the probe type, **port**, and the user-specified port number.

Note. See "Creating SLB Probes" on page 26-17 for a list of valid probe types.

For example, to set the TCP/UDP port for an HTTP SLB probe called "server_probe1" to 200 enter:

```
-> ip slb probe server_probe1 http port 200
```

Modifying the Probe Retries

By default, the number of SLB probe retries before deciding that a server is out of service is 3. To modify this value from 0 to 255 use the **ip slb probe retries** command by entering **ip slb probe** followed by the user-configured probe name, the probe type, **retries**, and the user-specified number of retries.

Note. See "Creating SLB Probes" on page 26-17 for a list of valid probe types.

For example, to set the number of retries for an HTTP SLB probe called "server_probe1" to 10, enter:

```
-> ip slb probe server_probe1 http retries 10
```

Configuring a Probe User Name

To configure a user name sent to a server as credentials for an HTTP GET operation to verify the health of the server use the **ip slb probe username** command by entering **ip slb probe** followed by the user-configured probe name, either **http** or **https**, **username**, and the user-specified user name.

For example, to set the user name for an HTTP SLB probe called "server_probe1" to "subnet1", enter:

```
-> ip slb probe server_probe1 http username subnet1
```

Configuring a Probe Password

To configure a password sent to a server as credentials for an HTTP GET to verify the health of the server use the **ip slb probe password** command by entering **ip slb probe** followed by the user-configured probe name, either **http** or **https**, **password**, and the user-specified password.

For example, to set the password for an HTTP SLB probe called "server_probe1" to "h1f45xc" enter:

```
-> ip slb probe server_probe1 http password h1f45xc
```

Configuring a Probe URL

To configure a URL sent to a server for an HTTP GET to verify the health of the server use the **ip slb probe url** command by entering **ip slb probe** followed by the user-configured probe name, either **http** or **https**, **url**, and the user-specified URL.

Note. The URL should be the relative web page name to be retrieved.

For example, to set the URL for an HTTP SLB probe called "server_probe1" to "pub/index.html", enter:

```
-> ip slb probe server_probe1 http url pub/index.html
```

Modifying the Probe Status

To modify this value, use the **ip slb probe status** command by entering **ip slb probe** followed by the user-configured probe name, either **http** or **https**, **status**, and the user-specified expected status.

For example, to set the expected status for an HTTP SLB probe called “server_probe1” to 404, enter:

```
-> ip slb probe server_probe1 http status 404
```

Configuring a Probe Send

To configure an ASCII string sent to a server to invoke a response from it and to verify its health use the **ip slb probe send** command by entering **ip slb probe** followed by the user-configured probe name, the valid probe type (**udp** or **tcp**), **send**, and the user-specified ASCII string.

For example, to set the TCP/UDP port for an TCP SLB probe called “server_probe1” to “test”, enter:

```
-> ip slb probe server_probe1 tcp send test
```

Configuring a Probe Expect

To configure an ASCII string used to compare a response from a server to verify the health of the server use the **ip slb probe expect** command by entering **ip slb probe** followed by the user-configured probe name, the valid probe type (**http**, **https**, **udp**, or **tcp**), **expect**, and the user-specified ASCII string.

For example, to set the TCP/UDP port for an HTTP SLB probe called “server_probe1” to “test”, enter:

```
-> ip slb probe server_probe1 http expect test
```

Displaying Server Load Balancing Status and Statistics

You can use CLI **show** commands to display the current configuration and statistics of Server Load Balancing on a switch. These commands include the following:

show ip slb	Displays the status of server load balancing on a switch.
show ip slb servers	Displays the status of all the physical servers belonging to server load balancing clusters on a switch.
show ip slb clusters	Displays the status and configuration of all server load balancing clusters on a switch. Also displays traffic statistics for all condition clusters.
show ip slb cluster	Displays detailed status and configuration information for a single server load balancing cluster on a switch. Also displays traffic statistics for a single condition cluster.
show ip slb cluster server	Displays detailed status and configuration information for a single physical server in a server load balancing cluster.
show ip slb probes	Display the configuration of Server Load Balancing (SLB) probes.

The **show ip slb**, **show ip slb servers**, and **show ip slb clusters** commands provide a "global" view of switch-wide SLB parameters. These commands are particularly helpful in fine-tuning configurations. For example, if you wanted to get a quick look at the status of all SLB clusters you would use the **show ip slb clusters** command as shown below:

```
-> show ip slb clusters
                                Admin    Operational          #     %
Cluster Name     VIP/COND       Status   Status               Srv   Avail
----------------+---------------+--------+--------------------+-----+---------
WorldWideWeb     128.241.130.204  Enabled  In Service         3     95
Intranet         128.241.130.210  Enabled  In Service         2     100
FileTransfer     128.241.130.206  Enabled  Out of Service     2     50
```

In the example above, two SLB clusters ("WorldWideWeb" and "Intranet") are administratively enabled and are "in service" (at least one physical server is operational in the cluster). The third SLB cluster ("FileTransfer") is administratively enabled but is "out of service" (no physical servers are operational in the cluster).

The **show ip slb cluster** command provides detailed configuration information and statistics for individual SLB clusters. To use the **show ip slb cluster** command, enter the command followed by the name of the SLB cluster, as shown below:

```
-> show ip slb cluster WorldWideWeb
```

A **statistics** parameter is available with both the **show ip slb clusters** and **show ip slb cluster** commands to provide a packet count of traffic that was qualified and sent to a QoS policy condition cluster. To use this parameter, enter either of these commands with their required parameters and optionally specify the statistics parameter, as shown below:

```
-> show ip slb clusters statistics
-> show ip slb cluster Intranet statistics
```

Note. See page 26-3 and page 26-5 for samples of the **show ip slb cluster** command output.

The **show ip slb cluster server** command provides detailed configuration information and statistics for individual SLB servers. To use the **show ip slb cluster server** command, enter the command, the name of the SLB cluster that the server belongs to, **server**, and the IP address of the server. For example, to display statistics and parameters for a server with an IP address of 10.123.11.14 that belongs to an SLB cluster called "Web_Server" you would enter:

```
-> show ip slb cluster Web_Server server 10.123.11.14
```

A screen similar to the following is displayed:

```
Cluster Web_Server
  VIP: 10.123.11.14
    Server 10.123.11.4
      Admin weight = 3,
      Admin status: Enabled,
      Oper status: In Service,
      Availability time (%)= 95,
      Ping failures = 0,
      Last ping round trip time (milliseconds)= 20,
      Probe status = OK
```

In the example above, the server with an IP address of 10.123.11.4 is shown to be administratively enabled and "in service" (this means that, this server is being used for SLB cluster client connections) with the administrative weight assigned as 3.

The **show ip slb probes** command provides both a global view of SLB probes and a detailed configuration information and statistics for individual probes. For example, to view the status of all probes enter show ip slb probes as shown below:

```
-> show ip slb probes

Probe Name              Period  Retries Timeout Method
----------------------+-------+-------+--------+------
web_server               60000       3    12000  HTTP
ail_server               60000       3     3000  SMTP
is_servers             3600000       5    24000  Ping
```

In the example above there are three probes configured on the switch.

To view detailed information on a single probe enter **show ip slb probes** followed by the probe name as shown in the example below:

```
-> show ip slb probes phttp
Probe phttp
  Type                          = HTTP,
  Period (seconds)              = 60,
  Timeout (milliseconds)        = 3000,
  Retries                       = 3,
  Port                          = 0,
  Username                      = ,
  Password                      = ,
  Expect                        = ,
  Status                        = 200,
  URL                           = /,
```

Note. See the "Server Load Balancing Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for complete syntax information on SLB **show** commands.

27 Configuring IP Multicast Switching

IP Multicast Switching is a one-to-many communication technique employed by emerging applications, such as video distribution, news feeds, conferencing, netcasting, and resource discovery (OSPF, RIP2, and BOOTP). Unlike unicast, which sends one packet per destination, multicast sends one packet to all devices in any subnetwork that has at least one device requesting the multicast traffic. Multicast switching also requires much less bandwidth than unicast techniques and broadcast techniques, since the source hosts only send one data stream to the ports on which destination hosts that request it are attached.

Destination hosts signal their intent to receive a specific IP multicast stream by sending a request to do so to a nearby switch by using Internet Group Management Protocol (IGMP). This is referred to as IGMP Snooping. Destination hosts signal their intent to receive a specific IPv6 multicast stream by sending a request to do so to a nearby switch by using Multicast Listener Discovery (MLD) protocol. This is referred to as MLD Snooping. The switch then learns on which ports multicast group subscribers are attached and can intelligently deliver traffic only to the respective ports. The OmniSwitch implementation of IGMP Snooping is called IP Multicast Switching (IPMS) and MLD snooping is called IP Multicast Switching version 6 (IPMSv6). IPMS/IPMSv6 allows switches to efficiently deliver multicast traffic in hardware at wire speed.

In This Chapter

This chapter describes the basic components of IPMS and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

This chapter includes the following topics:

- "IPMS Default Values" on page 27-2.
- "IPMSv6 Default Values" on page 27-3.
- "IPMS Overview" on page 27-4.
- "Interaction With Other Features" on page 27-7.
- "Configuring IPMS on a Switch" on page 27-10.
- "Modifying IPMS Parameters" on page 27-18.
- "IPMS Application Example" on page 27-45.
- "Displaying IPMS Configurations and Statistics" on page 27-49
- "IPMSv6 Overview" on page 27-28
- "Configuring IPMSv6 on a Switch" on page 27-30

- "Modifying IPMSv6 Parameters" on page 27-36.
- "IPMSv6 Application Example" on page 27-47.
- "Displaying IPMSv6 Configurations and Statistics" on page 27-50.

IPMS Default Values

The table below lists default values for the OmniSwitch IPMS implementation.

Parameter Description	Command	Default Value/Comments
Administrative Status	**ip multicast admin-state**	disabled
Flood initial unknown multicast traffic	**ip multicast flood-unknown**	disabled
IGMP Querier Forwarding	**ip multicast querier-forwarding**	disabled
IGMP Version	**ip multicast version**	version 2
IGMP Query Interval	**ip multicast query-interval**	125 seconds
IGMP Last Member Query Interval	**ip multicast last-member-query-interval**	10 tenths-of-seconds
IGMP Query Response Interval	**ip multicast query-response-interval**	100 tenths-of-seconds
IGMP Router Timeout	**ip multicast router-timeout**	90 seconds
Source Timeout	**ip multicast source-timeout**	30 seconds
IGMP Querying	**ip multicast querying**	disabled
IGMP Robustness	**ip multicast robustness**	2
IGMP Spoofing	**ip multicast spoofing**	disabled
IGMP Zapping	**ip multicast zapping**	disabled
Use an all-zero source IPv4 address for IGMP query packets	**ip multicast zero-based-query**	enabled

IPMSv6 Default Values

The table below lists default values for the OmniSwitch IPMSv6 implementation.

Parameter Description	Command	Default Value/Comments
Administrative Status	**ipv6 multicast admin-state**	disabled
Flood initial unknown multicast traffic	**ipv6 multicast flood-unknown**	disabled
MLD Querier Forwarding	**ipv6 multicast querier-forwarding**	disabled
MLD Version	**ipv6 multicast version**	version 1
MLD Query Interval	**ipv6 multicast query-interval**	125 seconds
MLD Last Member Query Interval	**ipv6 multicast last-member-query-interval**	1000 milliseconds
MLD Query Response Interval	**ipv6 multicast query-response-interval**	10000 milliseconds
MLD Router Timeout	**ipv6 multicast router-timeout**	90 seconds
Source Timeout	**ipv6 multicast source-timeout**	30 seconds
MLD Querying	**ipv6 multicast querying**	disabled
MLD Robustness	**ipv6 multicast robustness**	2
MLD Spoofing	**ipv6 multicast spoofing**	disabled
MLD Zapping	**ipv6 multicast zapping**	disabled
Use an all-zero source IPv6 address for MLD query packets	**ipv6 multicast zero-based-query**	enabled

IPMS Overview

A multicast group is defined by a multicast group address, which is a Class D IP address in the range 224.0.0.0 to 239.255.255.255. (Addresses in the range 239.0.0.0 to 239.255.255.255 are reserved for boundaries.) The multicast group address is indicated in the destination address field of the IP header. (See "Reserved IP Multicast Addresses" on page 27-5 for more information.)

IPMS tracks the source VLAN or the source Shortest Path Bridging (SPB) service on which the Internet Group Management Protocol (IGMP) requests are received. The network interfaces verify that a multicast packet is received by the switch on the source (or expected) port.

IPMS Example

The figure on the following page shows an IPMS network where video content can be provided to clients that request it. A server is attached to the switch that provides the source (i.e, multicast) IP addresses. Clients from two different attached networks send IGMP reports to the switch to receive the video content.

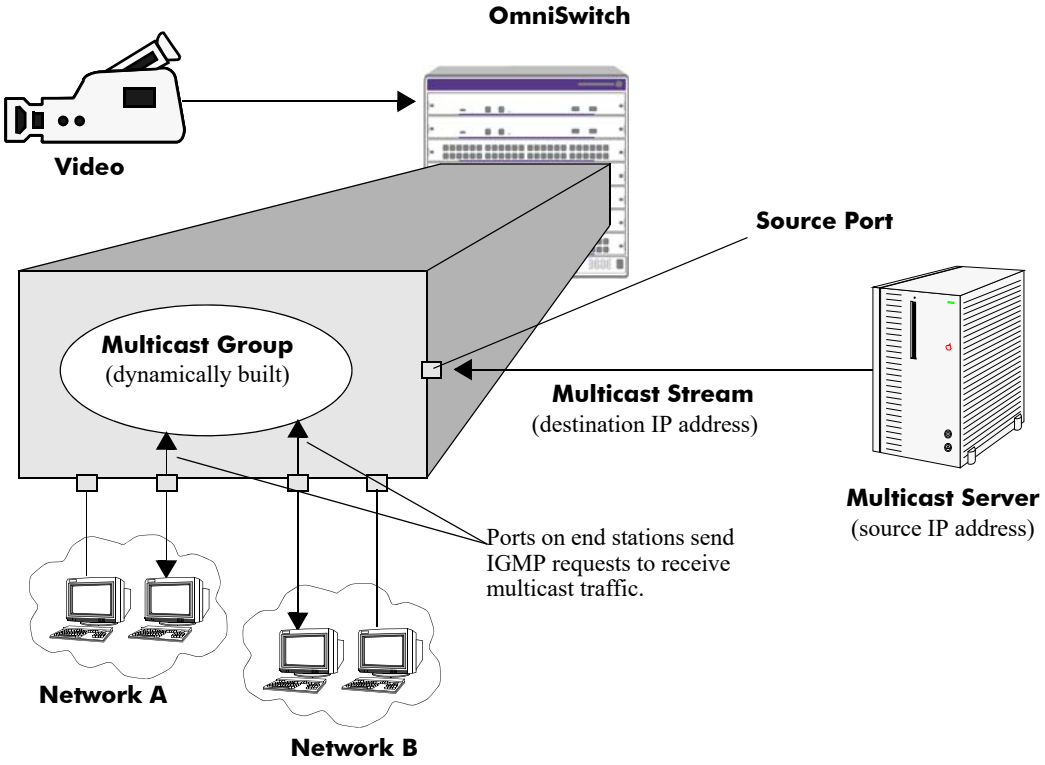


Figure 27-1 :Example of an IPMS Network

Reserved IP Multicast Addresses

The Internet Assigned Numbers Authority (IANA) created the range for multicast addresses, which is 224.0.0.0 to 239.255.255.255. However, as the table below shows, certain addresses are reserved and cannot be used.

Address or Address Range	Description
224.0.0.0 through 224.0.0.255	Routing protocols (e.g., OSPF, RIP2)
224.0.1.0 through 224.0.1.255	Internetwork Control Block (e.g., RSVP, DHCP, commercial servers)
224.0.2.0 through 224.0.255.0	AD-HOC Block (e.g., commercial servers)
224.1.0.0 through 224.1.255.255	ST Multicast Groups
224.2.0.0 through 224.2.255.255	SDP/SAP Block
224.252.0.0 through 224.255.255.255	DIS Transient Groups
225.0.0.0 through 231.255.255.255	Reserved
232.0.0.0 through 232.255.255.255	Source Specific Multicast
233.0.0.0 through 233.255.255.255	GLOP Block
234.0.0.0 through 238.255.255.255	Reserved
239.0.0.0 through 239.255.255.255	Administratively Scoped

IP Multicast Routing

IP multicast routing can be used for IP Multicast Switching and Routing (IPMSR). IP multicast routing is a way of controlling multicast traffic across networks. The IP multicast router discovers which networks want to receive multicast traffic by sending out Internet Group Management Protocol (IGMP) queries and receiving IGMP reports from attached networks. The IGMP reports signal that users want to join a multicast group.

If there is more than one IP multicast router in the network, the router with the lowest IP address is elected as the querier router, which is responsible for querying the subnetwork for group members.

The IP multicast routing package provides the following two separate protocols:

- Protocol Independent Multicast — Sparse Mode (PIM-SM) and Dense Mode (PIM-DM), which is described in "PIM" on page 27-6.
- Distance Vector Multicast Routing Protocol (DVMRP), which is described in "DVMRP" on page 27-6.

The multicast routing protocols build and maintain a multicast routing database. The multicast routing protocols forward multicast traffic to *networks* that have requested group membership to a specific multicast group. IPMS uses decisions made by the routing protocols and forwards multicast traffic to *ports* that request group membership. See the *OmniSwitch AOS Release 8 Advanced Routing Configuration Guide* for more information on IP multicast routing protocols.

PIM

Protocol-Independent Multicast (PIM) is an IP multicast routing protocol that uses routing information provided by unicast routing protocols, such as RIP and OSPF. Sparse Mode PIM (PIM-SM) contrasts with flood-and-prune dense mode multicast protocols, such as DVMRP and PIM Dense Mode (PIM-DM), in that multicast forwarding in PIM-SM is initiated only through specific requests. Downstream routers must explicitly join PIM-SM distribution trees in order to receive multicast streams on behalf of directly-connected receivers or other downstream PIM-SM routers. This paradigm of receiver-initiated forwarding makes PIM-SM ideal for network environments where receiver groups are thinly populated and bandwidth conservation is a concern, such as in Wide Area Networks (WANs). PIM-DM packets are transmitted on the same socket as PIM-SM packets as both use the same protocol and message format. Unlike PIM-SM, in PIM-DM there are no periodic joins transmitted; only explicitly triggered prunes and grafts. In PIM-DM, unlike PIM-SM, there is no Rendezvous Point (RP).

DVMRP

Distance Vector Multicast Routing Protocol (DVMRP) is a distributed multicast routing protocol that dynamically generates per-source delivery trees based upon routing exchanges. When a multicast source begins to transmit, the multicast data is flooded down the delivery tree to all points in the network. DVMRP then *prunes* (i.e., removes branches from) the delivery tree where the traffic is unwanted. This is in contrast to PIM-SM, which uses receiver-initiated (i.e., forward path) multicasting.

IGMP Version 3

IGMP is used by IPv4 systems (hosts and routers) to report their IP multicast group memberships to any neighboring multicast routers. IGMP Version 2 (IGMPv2) handles forwarding by IP multicast destination address only. IGMP Version 3 (IGMPv3) handles forwarding by source IP address and IP multicast destination address. All three versions (IGMPv1, IGMPv2, and IGMPv3) are supported.

Note. See "Configuring the IGMP Version" on page 27-13 for information on configuring the IGMP version.

In IGMPv2, each membership report contains only one multicast group. In IGMPv3, membership reports contain many multicast groups up to the Maximum Transmission Unit (MTU) size of the interface. IGMPv3 uses source filtering and reports multicast memberships to neighboring routers by sending membership reports. IGMPv3 also supports Source Specific Multicast (SSM) by allowing hosts to report interest in receiving packets only from specific source addresses or from all but specific source addresses.

Interaction With Other Features

This section contains important information about IP Multicast Switching (IPMS) interaction with other OmniSwitch features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

IPMS for Shortest Path Bridging

In a networking environment where IP multicast traffic is used, destination hosts signal their intent to receive a specific IP multicast stream by sending one of the following types of requests to a nearby switch:

- An Internet Group Management Protocol (IGMP) request to receive an IPv4 multicast stream.
- An Multicast Listener Discovery (MLD) protocol request to receive an IPv6 multicast stream.

The switch then learns on which ports multicast group subscribers are attached and can intelligently deliver traffic only to the respective ports. This process is referred to as IGMP or MLD snooping. The OmniSwitch implementation of IGMP and MLD snooping is called IP Multicast Switching (IPMS).

IPMS for Shortest Path Bridging (SPB) services is essentially the same. An SPB backbone edge bridge (BEB) will apply the logic of IPMS on a per-service basis to limit the traffic going out of each Service Access Point (SAP) port, as well as limit traffic going out across each backbone port. The SPB bridge will monitor the IGMP or MLD queries and requests from SAPs and Service Distribution Point (SDP) ports (also referred to as network virtual ports) to build the stream membership association logic and timing in the same manner as is done on a standard IGMP or MLD snooping bridge.

When traffic arrives on a SAP port, the switch will examine the packet to see if there are any known receivers. If there are any such receivers, then only ports (including network virtual ports) will have a copy of that frame sent on them. When traffic arrives from the core on a network virtual port, the same logic is applied so that a copy of the frame is only sent out on a port where a listener has requested membership to the stream. However, traffic from the core is never sent back into the core (split horizon protection).

The implementation of IPMS for SPB helps to cut down on the unnecessary forwarding of IP multicast traffic that can occur in an SPB network. This is particularly useful in networks that carry large amounts of multicast traffic, such as traffic from security cameras and on demand video.

Refer to the “Configuring Shortest Path Bridging” chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide* for more information about configuring SPB services.

VLAN and Service Domains

IPMS functionality is supported within the OmniSwitch VLAN domain and the OmniSwitch SPB service domain.

- The VLAN domain is identified by a VLAN ID. In the VLAN domain, each VLAN is accessed through a physical port. Each physical port can have more than one VLAN attached.
- The SPB service domain is identified by an SPB service instance identifier (I-SID), which is associated with an SPB service ID.

The following IPMS functionality is not supported in the SPB service domain:

- SSM Mapping
- Initial Packet Buffering
- Helper Address

Consider the following operational guidelines when configuring IPMS for SPB services:

- IPMS must be explicitly enabled for each SPB service; globally enabling IPMS for all services is not supported. However, globally disabling IPMS for all SPB services is supported.
- Configuration of IPMS has no effect on non-IP multicast traffic. However, if non-IP multicast traffic is sent with a destination MAC address that is reserved for IP multicast traffic and the forwarding mode is set to MAC address, IPMS may still end up processing this traffic when learning unregistered multicast sources. Once the multicast sources are learned, any matching traffic is forwarded the same way.
- Because the IPMS querying operation is not supported on services, an external querier is required to prevent group memberships from expiring.
- To help improve IPMS performance within the SPB service domain, consider the following recommendations:
  - Use IGMP version 3 (IGMPv3) and MLD version 2 (MLDv2).
  - Enable the IPMS fast join function to help accelerate convergence and reduce network flooding.
  - Increase the source timeout for snooping enabled services to help reduce control plane overhead.
- Using IPMS with wildcard SAP ports is not recommended.For example, 1/1/10:all, where "all" specifies a wildcard SAP encapsulation of arbitrary customer VLANs (CVLAN). When a last member query (group-specific or group-source-specific) is received, the last member query timeout processing of memberships learned on a wildcard SAP only occurs if the query is untagged.

Consider the following guidelines when configuring IPMS for SPB services on an OmniSwitch 6860, OmniSwitch 6865, and OmniSwitch 6900:

- When IPMS is enabled on at least one SPB service, the control plane begins processing unknown IP multicast traffic from all services. This is the only supported flood unknown behavior for services.
  - Flooding only occurs when the control plane software is initially learning and registering new sources in the Layer 2 forwarding database (FDB).
  - IP multicast sources from services on which IPMS is enabled are forwarded to SDPs and SAPs on which multicast routers or clients are present.
  - IP multicast traffic from any service that does not have IPMS enabled, is flooded to all SDPs and SAPs that are associated with the service.
  - This process has no effect on IGMP or MLD protocol traffic.
  - This process occurs independently for IPv4 and IPv6 multicast.
- IP multicast snooping for services does not snoop MAC addresses that fall within the range of 01:00:5e:00:00:00/40 and 33:33:00:00:00:00/40 and the traffic is allowed to flood even if the flooding of unknown multicast traffic is disabled. Avoid using any multicast groups that map to the excluded MAC addresses.
- Ingress flood rate limiting of IP multicast traffic is diminished when IPMS is used for SPB services.
  - Ingress flood rate limiting of IP multicast traffic only applies when the control plane is learning and registering new sources in the FDB.
  - Do not configure IPMS for SPB services if this functionality is required.

Consider the following guidelines when configuring IPMS for SPB services on an OmniSwitch 9900:

- IPv4 multicast snooping for services does not snoop 224.0.0.0/24 and the traffic is allowed to flood even if the flooding of unknown multicast traffic is disabled. Avoid using any multicast groups that map to these excluded IPv4 addresses.
- IPv6 multicast snooping for VLANs or services does not snoop ff02::/120 and the traffic is allowed to flood even if the flooding of unknown multicast traffic is disabled. Avoid using any multicast groups that map to these excluded IPv6 addresses.
- Using IPMS only on SPB services that are configured to perform VLAN translation is recommended on all switches in the network to ensure full communication.
  - When IPMS is configured for an SPB service on an OmniSwitch 9900, VLAN translation is implicitly enabled for the service, even if the configured VLAN translation status is disabled for the service. The VLAN translation status is no longer configurable as long as IPMS is enabled for the service.
  - Mixing switches with VLAN translation enabled on some and disabled on other switches in the same network is not recommended. Make sure all switches have VLAN translation enabled, especially if an OmniSwitch 9900 is added to the network.

Refer to the “Service Manager Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* and the “Configuring Shortest Path Bridging” chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide* for more information about configuring SPB service components.

Configuring IPMS on a Switch

This section describes how to use Command Line Interface (CLI) commands to complete the following configuration tasks:

- Enable and disable IP Multicast Switching and Routing (IPMSR) switch wide (see "Enabling and Disabling IP Multicast Status" on page 27-10).
- Configure a port as a IGMP static neighbor (see "Configuring and Removing an IGMP Static Neighbor" on page 27-13).
- Configure a port as a IGMP static querier (see "Configuring and Removing an IGMP Static Querier" on page 27-14).
- Configure a port as a IGMP static group (see "Configuring and Removing an IGMP Static Group" on page 27-15).

In addition, a tutorial is provided in "IPMS Application Example" on page 27-45 that shows how to use CLI commands to configure a sample network.

Note. See the "IP Multicast Switching Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for complete documentation of IPMS CLI commands.

Enabling and Disabling IP Multicast Status

IP Multicast Switching and Routing is disabled by default on a switch. The following subsections describe how to enable and disable IP Multicast Switching and Routing with the **ip multicast admin-state** command.

Note. If IP Multicast switching and routing is enabled on the system, the VLAN or Shortest Page Bridging (SPB) service configuration overrides the configuration of the system.

Enabling IP Multicast Status

To enable IP Multicast switching and routing on the system if no VLAN or SPB service is specified, use the **ip multicast admin-state** command as shown below:

```
-> ip multicast admin-state enable
```

You can also enable IP Multicast switching and routing on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 admin-state enable
-> ip multicast service 10 admin-state enable
```

Disabling IP Multicast Status

To disable IP Multicast switching and routing on the system if no VLAN or SPB service is specified, use the **ip multicast admin-state** command as shown below:

```
-> ip multicast admin-state disable
```

Or, as an alternative, enter the following command to restore the IP Multicast status to its default setting:

```
-> no ip multicast admin-state
```

You can also disable IP Multicast switching and routing on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 admin-state disable
-> ip multicast service 10 admin-state disable
```

Or, as an alternative, enter the following commands to restore the IP Multicast status to its default setting:

```
-> no ip multicast vlan 2 admin-state
-> no ip multicast service 10 admin-state
```

Enabling and Disabling Flooding of Unknown Multicast Traffic

When a traffic flow is first seen on a port. there is a brief period of time where traffic may get dropped before the forwarding information is calculated. When flooding unknown multicast traffic is enabled, no packets are dropped before the forwarding information is available.

By default, the flooding of unknown multicast traffic is disabled.The following subsections describe how to enable and disable this function by using the **ip multicast flood-unknown** command.

Enabling Flooding of Unknown Multicast Traffic

You can enable the flooding of unknown multicast traffic on the system by entering **ip multicast flood-unknown** followed by the **enable** keyword. For example:

```
-> ip multicast flood-unknown enable
```

You can also enable the flooding of unknown multicast traffic on a specific VLAN or SPB service by entering:

```
-> ip multicast vlan 2 flood-unknown enable
-> ip multicast service 10 flood-unknown enable
```

Disabling Flooding of Unknown Multicast Traffic

You can disable the flooding of unknown multicast traffic on the system by entering **ip multicast flood-unknown** followed by the **disable** keyword. For example:

```
-> ip multicast flood-unknown disable
```

Or, as an alternative, enter the following command to restore the flooding of unknown multicast traffic to its default setting:

```
-> no ip multicast flood-unknown
```

You can also disable the flooding of unknown multicast traffic on the specified VLAN or SPB service by entering**:**

```
-> ip multicast vlan 2 flood-unknown disable
-> ip multicast service 10 flood-unknown disable
```

Or, as an alternative, enter the following commands to restore the flooding of unknown multicast traffic to its default setting:

```
-> no ip multicast vlan 2 flood-unknown
-> no ip multicast service 10 flood-unknown
```

Enabling and Disabling IGMP Querier-forwarding

By default, IGMP querier-forwarding is disabled.The following subsections describe how to enable and disable IGMP querier-forwarding by using the **ip multicast querier-forwarding** command.

Enabling the IGMP Querier-forwarding

You can enable the IGMP querier-forwarding by entering **ip multicast querier-forwarding** followed by the **enable** keyword. For example, to enable the IGMP querier-forwarding on the system if no VLAN or SPB service is specified, you would enter:

```
-> ip multicast querier-forwarding enable
```

You can also enable the IGMP querier-forwarding on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 querier-forwarding enable
-> ip multicast service 10 querier-forwarding enable
```

Disabling the IGMP Querier-forwarding

You can disable the IGMP querier-forwarding by entering **ip multicast querier-forwarding** followed by the **disable** keyword. For example, to disable the IGMP querier-forwarding on the system if no VLAN or SPB service is specified, you would enter:

```
-> ip multicast querier-forwarding disable
```

Or, as an alternative, enter the following command to restore the IGMP querier-forwarding to its default setting:

```
-> no ip multicast querier-forwarding
```

You can also disable the IGMP querier-forwarding on the specified VLAN or SPB service by entering**:**

```
-> ip multicast vlan 2 querier-forwarding disable
-> ip multicast service 10 querier-forwarding disable
```

Or, as an alternative, enter the following commands to restore the IGMP querier-forwarding to its default setting:

```
-> no ip multicast vlan 2 querier-forwarding
-> no ip multicast service 10 querier-forwarding
```

Configuring and Restoring the IGMP Version

By default, the version of Internet Group Management Protocol (IGMP) membership is Version 2. The following subsections describe how to configure IGMP protocol version ranging from 1 to 3 with the **ip multicast version** command.

Configuring the IGMP Version

To change the IGMP protocol version on the system if no VLAN or SPB service is specified, use the **ip multicast version** command as shown below:

```
-> ip multicast version 3
```

You can also change the IGMP protocol version on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 5 version 1
-> ip multicast service 10 version 1
```

Restoring the IGMP Version

To restore the IGMP protocol version to its default version on the system if no VLAN or SPB service is specified, use the **ip multicast version** command as shown below:

```
-> ip multicast version 0
```

Or, as an alternative, enter the following command to restore the IGMP version to its default version:

```
-> no ip multicast version
```

You can also restore the IGMP protocol version to version 2 on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 version 0
-> ip multicast service 10 version 0
```

Or, as an alternative, enter the following commands to restore the IGMP version to its default version:

```
-> no ip multicast vlan 2 version
-> no ip multicast service 10 version
```

Configuring and Removing an IGMP Static Neighbor

IGMP static neighbor ports receive all multicast streams on the designated VLAN or SPB service and also receive IGMP reports for the VLAN or SPB service. The following subsections describe how to configure and remove an IGMP static neighbor port by using the **ip multicast static-neighbor** command.

Configuring an IGMP Static Neighbor

To configure a port as an IGMP static neighbor port, use the **ip multicast static-neighbor** command with the **port** option. For example, the following command configures port 1/1/13 in VLAN 2 as an IGMP static neighbor:

```
-> ip multicast static-neighbor vlan 2 port 1/1/13
```

To configure an SPB Service Access Point (SAP) as an IGMP static neighbor port, use the **ip multicast static-neighbor** command with the **sap port** option. For example, the following command configures SAP port 1/1/23:10 that is bound to SPB service 10 as an IGMP static neighbor:

```
-> ip multicast static-neighbor service 10 sap port 1/1/23:10
```

In this example, 1/1/23:10 serves as a SAP ID, which is comprised of an access port number (1/1/23) and an encapsulation value (10). SPB service 10 is mapped to SAP ID 1/1/23:10. Traffic received on access port 1/1/23 that is tagged with VLAN 10 is encapsulated and then forwarded on service 10 through the SPB network. Refer to Chapter 10, "Service Manager Commands," for more information.

To create an IGMP static neighbor entry on a link aggregate, use the **linkagg** parameter. For example:

```
-> ip multicast static-neighbor vlan 2 linkagg 7
-> ip multicast static-neighbor service 10 sap linkagg 10:100
```

Removing an IGMP Static Neighbor

To reset the port so that it is no longer an IGMP static neighbor port, use the **no** form of the **ip multicast static-neighbor** command with the **vlan** and **port** parameters. For example, the following command removes port 1/1/13 with designated VLAN 2 as an IGMP static neighbor:

```
-> no ip multicast static-neighbor vlan 2 port 1/1/13
```

To reset the SAP port so that it is no longer an IGMP static neighbor port, use the **no** form of the **ip multicast static-neighbor** command with the **service** and **sap port** parameters. For example, the following command removes SAP port 1/1/23 with designated service 10 as an IGMP static neighbor:

```
-> no ip multicast static-neighbor service 10 sap port 1/1/23
```

Configuring and Removing an IGMP Static Querier

IGMP static querier ports receive IGMP reports generated on the designated VLAN or SPB service. Unlike IPMS neighbor ports, they do not receive all multicast streams. The following subsections describe how to configure and remove a static querier by using the **ip multicast static-querier** command.

Configuring an IGMP Static Querier

To configure a port as an IGMP static querier port, use the **ip multicast static-neighbor** command with the **port** option. For example, the following command configures port 1/1/13 in VLAN 2 as an IGMP static neighbor:

```
-> ip multicast static-querier vlan 2 port 1/1/13
```

To configure an SPB Service Access Point (SAP) as an IGMP static querier port, use the **ip multicast static-querier** command with the **sap port** option. For example, the following command configures SAP port 1/1/23:10 that is bound to SPB service 10 as an IGMP static querier:

```
-> ip multicast static-querier service 10 sap port 1/1/23:10
```

In this example, 1/1/23:10 serves as a SAP ID, which is comprised of an access port number (1/1/23) and an encapsulation value (10). SPB service 10 is mapped to SAP ID 1/1/23:10. Traffic received on access port 1/1/23 that is tagged with VLAN 10 is encapsulated and then forwarded on service 10 through the SPB network. Refer to Chapter 10, "Service Manager Commands," for more information.

To create an IGMP static querier entry on a link aggregate, use the **linkagg** parameter. For example:

```
-> ip multicast static-querier vlan 2 linkagg 7
-> ip multicast static-querier service 10 sap linkagg 10:100
```

Removing an IGMP Static Querier

To reset the port so that it is no longer an IGMP static querier port, use the **no** form of the **ip multicast static-querier** command. For example, the following command removes port 1/1/13 with designated VLAN 2 as an IGMP static querier:

```
-> no ip multicast static-querier vlan 2 port 1/1/13
```

To reset the SAP port so that it is no longer an IGMP static querier port, use the **no** form of the **ip multicast static-querier** command with the **service** and **sap port** parameters. For example, the following command removes SAP port 1/1/23 with designated service 10 as an IGMP static querier:

```
-> no ip multicast static-querier service 10 sap port 1/1/23
```

Configuring and Removing an IGMP Static Group

IGMP static group ports receive the multicast streams generated for the specified IP Multicast group address. The following subsections describe how to configure and remove a static group with the **ip multicast static-group** command.

Configuring an IGMP Static Group

To configure an IGMP static group entry on a port, use the **ip multicast static-group** command with the **port** option. For example, the following command configures an IGMP static group entry with an IP address of 225.0.0.1 on port 1/1/3 in VLAN 3:

```
-> ip multicast static-group 225.0.0.1 vlan 3 port 1/1/13
```

To configure an IGMP static group entry on an SPB Service Access Point (SAP), use the **ip multicast static-group** command with the **sap port** option. For example, the following command configures an IGMP static group entry with IP address 225.0.0.1 on SAP port 1/1/23:10 that is bound to SPB service 10:

```
-> ip multicast static-group 225.0.0.1 service 10 sap port 1/1/23:10
```

In this example, 1/1/23:10 serves as a SAP ID, which is comprised of an access port number (1/1/23) and an encapsulation value (10). SPB service 10 is mapped to SAP ID 1/1/23:10. Traffic received on access port 1/1/23 that is tagged with VLAN 10 is encapsulated and then forwarded on service 10 through the SPB network. Refer to Chapter 10, “Service Manager Commands,” for more information.

To create an IGMP static group entry on a link aggregate, use the **linkagg** parameter. For example:

```
-> ip multicast static-group 225.0.0.1 vlan 2 linkagg 7
-> ip multicast static-group 225.0.0.1 service 10 sap linkagg 10:100
```

Removing an IGMP Static Group

To remove an IGMP static group entry from a port, use the **no** form of the **ip multicast static-group** command with the **vlan** and **port** parameters. For example, the following command removes the IGMP static group entry with an IP address of 225.0.0.1 on port 1/1/13:

```
-> no ip multicast static-group 225.0.0.1 vlan 3 port 1/1/13
```

To remove an IGMP static group entry from a SAP port, use the **no** form of the **ip multicast static-group** command with the **service** and **sap port** parameters. For example, the following command removes the IGMP static group entry with an IP address of 225.0.0.1 on SAP port 1/1/23:

```
-> no ip multicast static-group 225.0.0.1 service 10 sap port 1/1/23
```

Initial Multicast Packet Buffering

Multicast is often used for audio\video streaming applications, where the first packet is dropped as it is used for learning the new flow. However, if multicast is used for signaling applications, the initial packets sent by the multicast source are important. The packet buffering functionality can be enabled to prevent loss of first multicast packets in a routing environment.

Enabling Packet Buffering

To enable buffering of IPv4 and IPv6 initial multicast packets and support first packet routing, use the **ip multicast initial-packet-buffer admin-state** and **ipv6 multicast initial-packet-buffer admin-state** commands, respectively.

For example,

```
-> ip multicast initial-packet-buffer admin-state enable
-> ipv6 multicast initial-packet-buffer admin-state enable
```

All the initial multicast packets sent by the multicast source are buffered. The following are the limitations:

- In a given instance, the IPMS system can buffer only a configurable number of multicast packets per flow that are trapped to the IPMS software for learning the multicast flow. If the number of packets buffered for a given flow reaches the maximum configured limit, then the IPMS system will not buffer any new multicast packets for that flow. The maximum number of initial multicast packets allowed to be buffered per multicast flow is configured using **ip multicast initial-packet-buffer max-packet** and **ipv6 multicast initial-packet-buffer max-packet** for IPv4 and IPv6, respectively.

  For example:

  ```
  -> ip multicast initial-packet-buffer max-packet 4
  -> ipv6 multicast initial-packet-buffer max-packet 4
  ```

- In a given instance, the IPMS system can buffer initial multicast packets only for a configurable number of flows globally on the switch. If the number of flows that are being buffered reaches the maximum configured limit, then the IPMS will not buffer initial multicast packets for any new flows in the system. The maximum number of IPv4 and IPv6 multicast flows allowed for initial packet buffering is configured using **ip multicast initial-packet-buffer max-flow** and **ipv6 multicast initial-packet-buffer max-flow**, respectively.

  For example:

  ```
  -> ip multicast initial-packet-buffer max-flow 32
  -> ipv6 multicast initial-packet-buffer max-flow 32
  ```

- The buffered initial multicast packets for a flow will not be retained in the IPMS system forever. If the buffered packets are not sent to the destination (multicast clients) within a specified time, then the buffered packets will be dropped in the IPMS system. The timeout value for the initial buffered IPv4 and IPv6 multicast packets is configured using **ip multicast initial-packet-buffer timeout** and **ipv6 multicast initial-packet-buffer timeout**, respectively.

  For example:

  ```
  -> ip multicast initial-packet-buffer timeout 2
  -> ipv6 multicast initial-packet-buffer timeout 2
  ```

- There may be loss of a few initial multicast packets while programming the IPMS index and sending buffered packets;
  - When the multicast routing protocol has greater than 255 interfaces - In a given instance, if there are more that 255 interfaces, the IPMS system will initially send the buffered multicast packets to the clients connected to a maximum of 255 interfaces. Then the remaining route information will be processed. However, the IPMS system begins communicating with the NI without waiting for complete route information. This may cause a loss of buffered packets to the remaining interfaces.
  - When ingress and egress are on the same VLAN - In any instance, if local clients are present in the ingress VLAN, the IPMS system will immediately generate an IPMC index for the switching port and will send the buffered packets. However, there is a delay in receiving the response from the routed clients. This may cause a loss of buffered packets for the routed clients.

  These limitations are overcome by delaying the IPMC index generation in IPMS system. This delay is configured using **ip multicast initial-packet-buffer min-delay** and **ipv6 multicast initial-packet-buffer min-delay** for IPv4 and IPv6 multicast packets, respectively.

  ```
  -> ip multicast initial-packet-buffer min-delay 200
  -> ipv6 multicast initial-packet-buffer min-delay 200
  ```

- Flood-unknown and buffer packet features are mutually exclusive. Flood-unknown must be disabled for the packet buffering feature to function. Use **ip multicast flood-unknown** and **ipv6 multicast flood-unknown** commands to disable the flood-unknown feature for IPv4 and IPv6 multicast flow, respectively.

  For example:

  ```
  -> ip multicast flood-unknown disable
  -> ipv6 multicast flood-unknown disable
  ```

- Use **show configuration snapshot ipms**, **show ip multicast initial-packet-buffer** and **show ipv6 multicast initial-packet-buffer** commands to view the status of the packet buffering functionality.

Disabling Packet Buffering

To disable buffering of IPv4 initial multicast packet in the IPMS NI, use the following command. For example,

```
-> ip multicast initial-packet-buffer admin-state disable
```

To disable buffering of IPv6 initial multicast packet in the IPMS NI, use the following command. For example,

```
-> ipv6 multicast initial-packet-buffer admin-state disable
```

Modifying IPMS Parameters

The table in "IPMS Default Values" on page 27-2 lists default values for IPMS parameters. The following sections describe how to use CLI commands to modify these parameters.

Modifying the IGMP Query Interval

The default IGMP query interval (i.e., the time between IGMP queries) is 125 seconds. The following subsections describe how to configure a user-specified query interval value and restore it with the **ip multicast query-interval** command.

Configuring the IGMP Query Interval

You can modify the IGMP query interval from 1 to 65535 seconds by entering **ip multicast query-interval** followed by the new value. For example, to set the query interval to 60 seconds on the system if no VLAN or SPB service is specified, you would enter:

```
-> ip multicast query-interval 60
```

You can also modify the IGMP query interval on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 query-interval 60
-> ip multicast service 10 query-interval 60
```

Restoring the IGMP Query Interval

To restore the IGMP query interval to its default value on the system if no VLAN or SPB service is specified, use the **ip multicast query-interval** command by entering:

```
-> ip multicast query-interval 0
```

Or, as an alternative, enter the following command to restore the IGMP query interval to its default value:

```
-> no ip multicast query-interval
```

You can also restore the IGMP query interval to its default value on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 query-interval 0
-> ip multicast service 10 query-interval 0
```

Or, as an alternative, enter the following commands to restore the IGMP query interval to its default value:

```
-> no ip multicast vlan 2 query-interval
-> no ip multicast service 10 query-interval
```

Modifying the IGMP Last Member Query Interval

The default IGMP last member query interval (i.e., the time period to reply to an IGMP query message sent in response to a leave group message) is 10 tenths of seconds. The following subsections describe how to configure the IGMP last member query interval and restore it by using the **ip multicast last-member-query-interval** command.

Configuring the IGMP Last Member Query Interval

You can modify the IGMP last member query interval from 1 to 65535 tenths of seconds by entering **ip multicast last-member-query-interval** followed by the new value. For example, to set the IGMP last member query interval to 60 tenths-of-seconds on the system if no VLAN or SPB service is specified, you would enter:

```
-> ip multicast last-member-query-interval 60
```

You can also modify the IGMP last member query interval on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 3 last-member-query-interval 60
-> ip multicast service 10 last-member-query-interval 60
```

Restoring the IGMP Last Member Query Interval

To restore the IGMP last member query interval to its default value on the system if no VLAN or SPB service is specified, use the **ip multicast last-member-query-interval** command by entering:

```
-> ip multicast last-member-query-interval 0
```

Or, as an alternative, enter the following command to restore the IGMP last member query interval to its default value:

```
-> no ip multicast last-member-query-interval
```

You can also restore the IGMP last member query interval on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 last-member-query-interval 0
-> ip multicast service 10 last-member-query-interval 0
```

Or, as an alternative, enter the following commands to restore the IGMP last member query interval to its default value:

```
-> no ip multicast vlan 2 last-member-query-interval
-> no ip multicast service 10 last-member-query-interval
```

Modifying the IGMP Query Response Interval

The default IGMP query response interval (i.e., the time period to reply to an IGMP query message) is 100 in tenths of seconds. The following subsections describe how to configure the query response interval and how to restore it with the **ip multicast query-response-interval** command.

Configuring the IGMP Query Response Interval

You can modify the IGMP query response interval from 1 to 65535 tenths of seconds by entering **ip multicast query-response-interval** followed by the new value. For example, to set the IGMP query response interval to 6000 tenths-of-seconds on the system if no VLAN or SPB service is specified, you would enter:

```
-> ip multicast query-response-interval 6000
```

You can also modify the IGMP query response interval on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 3 query-response-interval 6000
-> ip multicast service 10 query-response interval 6000
```

Restoring the IGMP Query Response Interval

To restore the IGMP query response interval to its default value on the system if no VLAN or SPB service is specified, use the **ip multicast query-response-interval** command by entering:

```
-> ip multicast query-response-interval 0
```

Or, as an alternative, enter the following command to restore the IGMP query response interval to its default value:

```
-> no ip multicast query-response-interval
```

You can also restore the IGMP query response interval on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 query-response-interval 0
-> ip multicast service 10 query-response-interval 0
```

Or, as an alternative, enter the following commands to restore the IGMP query response interval to its default value:

```
-> no ip multicast vlan 2 query-response-interval
-> no ip multicast service 10 query-response-interval
```

Enabling and Disabling Zero-based IGMP Query

By default, an all-zero source IPv4 address is used for IGMP query packets when a non-querier is querying the membership of a port. The following subsections describe how to enable and disable using a zero-based IGMP query by using the **ip multicast zero-based-query** command.

Enabling Zero-based IGMP Query

You can enable the use of a zero-based IGMP query on the system by entering **ip multicast zero-based-query** followed by the **enable** keyword. For example:

```
-> ip multicast zero-based-query enable
```

Or, as an alternative, enter the following command to restore the zero-based IGMP query to its default setting (enabled):

```
-> no ip multicast zero-based-query
```

You can also enable the use of a zero-based IGMP query on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 zero-based-query enable
-> ip multicast service 10 zero-based-query enable
```

Or, as an alternative, enter the following command to restore the zero-based IGMP query to its default setting (enabled) for the specified VLAN or SPB service:

```
-> no ip multicast vlan 2 zero-based-query
-> no ip multicast service 10 zero-based-query
```

Disabling Zero-based IGMP Query

You can disable the use of a zero-based IGMP query on the system by entering **ip multicast zero-based-query** followed by the **disable** keyword. For example:

```
-> ip multicast zero-based-query disable
```

You can also disable the use of a zero-based IGMP query on the specified VLAN or SPB service by entering**:**

```
-> ip multicast vlan 2 zero-based-query disable
-> ip multicast service 10 zero-based-query disable
```

Modifying the IGMP Router Timeout

The default IGMP router timeout (i.e., expiry time of IP multicast routers) is 90 seconds. The following subsections describe how to configure a user-specified router timeout value and how to restore it with the **ip multicast router-timeout** command.

Configuring the IGMP Router Timeout

You can modify the IGMP router timeout from 1 to 65535 seconds by entering **ip multicast router-timeout** followed by the new value. For example, to set the IGMP router timeout to 360 seconds on the system if no VLAN or SPB service is specified, you would enter:

```
-> ip multicast router-timeout 360
```

You can also modify the IGMP router timeout on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 router-timeout 360
-> ip multicast service 10 router-timeout 360
```

Restoring the IGMP Router Timeout

To restore the IGMP router timeout to its default value on the system if no VLAN or SPB service is specified, use the **ip multicast router-timeout** command by entering:

```
-> ip multicast router-timeout 0
```

Or, as an alternative, enter the following command to restore the IGMP router timeout to its default value:

```
-> no ip multicast router-timeout
```

You can also restore the IGMP router timeout on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 router-timeout 0
-> ip multicast service 10 router-timeout 0
```

Or, as an alternative, enter the following commands to restore the IGMP router timeout to its default value:

```
-> no ip multicast vlan 2 router-timeout
-> no ip multicast service 10 router-timeout
```

Modifying the Source Timeout

The default source timeout (i.e., the expiry time of IP multicast sources) is 30 seconds. The following subsections describe how to configure a user-specified source timeout value and restore it by using the **ip multicast source-timeout** command.

Configuring the Source Timeout

You can modify the source timeout from 1 to 65535 seconds by entering **ip multicast source-timeout** followed by the new value. For example, to set the source timeout to 360 seconds on the system if no VLAN or SPB service is specified, you would enter:

```
-> ip multicast source-timeout 360
```

You can also modify the source timeout on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 source-timeout 360
-> ip multicast service 10 source-timeout 360
```

Restoring the Source Timeout

To restore the source timeout to its default value on the system if no VLAN or SPB service is specified, use the **ip multicast source-timeout** command by entering:

```
-> ip multicast source-timeout 0
```

Or, as an alternative, enter the following command to restore the source timeout to its default value:

```
-> no ip multicast source-timeout
```

You can also restore the source timeout on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 source-timeout 0
-> ip multicast service 10 source-timeout 0
```

Or, as an alternative, enter the following command to restore the source timeout to its default value:

```
-> no ip multicast vlan 2 source-timeout
-> no ip multicast service 10 source-timeout
```

Enabling and Disabling IGMP Querying

By default, IGMP querying is disabled.The following subsections describe how to enable and disable IGMP querying by using the **ip multicast querying** command.

Enabling the IGMP Querying

You can enable the IGMP querying by entering **ip multicast querying** followed by the **enable** keyword. For example, to enable the IGMP querying on the system if no VLAN is specified, you would enter:

```
-> ip multicast querying enable
```

You can also enable the IGMP querying on the specified VLAN by entering:

```
-> ip multicast vlan 2 querying enable
```

Specifying a Static Source IP Address

By default, a source IP address is not specified when IGMP querying is enabled; the system automatically determines the addresses to use for IGMP queries. However, a static source IP address can be specified to overcome the need for an IP interface. If configured, the static source IP is then always used for querying, regardless of the IP interface address or administrative state.

To configure a static source IP address for the system, use the **ip multicast querying** command with the **static-source-ip** parameter. For example:

```
-> ip multicast querying static-source-ip 10.2.2.1
```

To configure a static source IP address for a specific VLAN, use the **ip multicast querying** command with the **vlan** and **static-source-ip** parameters. For example:

```
-> ip multicast querying vlan 2 static-source-ip 10.2.2.1
```

To remove a static source IP address for the system or a specific VLAN, use the **no** form of the **ip multicast querying** command. For example:

```
-> no ip multicast querying static-source-ip
-> no ip multicast querying vlan 2 static-source-ip
```

Disabling the IGMP Querying

You can disable the IGMP querying by entering **ip multicast querying** followed by the **disable** keyword. For example, to disable the IGMP querying on the system if no VLAN is specified, you would enter:

```
-> ip multicast querying disable
```

Or, as an alternative, enter the following command to restore the IGMP querying to its default setting:

```
-> no ip multicast querying
```

You can also disable the IGMP querying on the specified VLAN by entering**:**

```
-> ip multicast vlan 2 querying disable
```

Or, as an alternative, enter the following command to restore IGMP querying to its default setting:

```
-> no ip multicast vlan 2 querying
```

Modifying the IGMP Robustness Variable

The default value of the IGMP robustness variable (i.e., the variable that allows fine-tuning on a network, where the expected packet loss is higher) is 2. The following subsections describe how to set the value of the robustness variable and restore it with the **ip multicast robustness** command.

Configuring the IGMP Robustness variable

You can modify the IGMP robustness variable from 1 to 7 on the system if no VLAN or SPB service is specified, by entering **ip multicast robustness** followed by the new value. For example, to set the value of IGMP robustness to 3 you would enter:

```
-> ip multicast robustness 3
```

Note. If the links are known to be lossy, then the robustness variable can be set to a higher value (7).

You can also modify the IGMP robustness variable from 1 to 7 on the specified VLAN or SPB service by entering**:**

```
-> ip multicast vlan 2 robustness 3
-> ip multicast service 10 robustness 3
```

Restoring the IGMP Robustness Variable

You can restore the IGMP robustness variable to its default value on the system if no VLAN or SPB service is specified, by entering **ip multicast robustness** followed by the value 0 as shown below:

```
-> ip multicast robustness 0
```

Or, as an alternative, enter the following command to restore the IGMP robustness to its default value:

```
-> no ip multicast robustness
```

You can also restore the IGMP robustness variable to its default value on the specified VLAN or SPB service by entering **ip multicast robustness** followed by the value 0 as shown below:

```
-> ip multicast vlan 2 robustness 0
-> ip multicast service 10 robustness 0
```

Or, as an alternative, enter the following commands To restore the IGMP robustness to its default value:

```
-> no ip multicast vlan 2 robustness
-> no ip multicast service 10 robustness
```

Enabling and Disabling the IGMP Spoofing

By default, IGMP spoofing (i.e., replacing a client's MAC and IP address with the system's MAC and IP address, when proxying aggregated IGMP group membership information) is disabled on the switch. The following subsections describe how to enable and disable spoofing by using the **ip multicast spoofing** command.

Enabling the IGMP Spoofing

To enable IGMP spoofing on the system if no VLAN or SPB service is specified, use the **ip multicast spoofing** command as shown below:

```
-> ip multicast spoofing enable
```

You can also enable IGMP spoofing on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 spoofing enable
-> ip multicast service 10 spoofing enable
```

Specifying a Static Source IP Address

By default, a source IP address is not specified when IGMP spoofing is enabled; the system automatically determines the addresses to use for spoofing. However, a static source IP address can be specified to overcome the need for an IP interface. If configured, the static source IP is then always used for spoofing, regardless of the IP interface address or administrative state.

To configure a static source IP address for the system, use the **ip multicast spoofing** command with the **static-source-ip** parameter. For example:

```
-> ip multicast spoofing static-source-ip 10.2.2.1
```

To configure a static source IP address for a specific VLAN or SPB service, use the **ip multicast spoofing** command with the **vlan** and **static-source-ip** parameters. For example:

```
-> ip multicast vlan 2 spoofing static-source-ip 10.2.2.1
-> ip multicast service 10 spoofing static-source-ip 10.2.2.1
```

To remove a static source IP address for the system or a specific VLAN or SPB service, use the **no** form of the **ip multicast spoofing** command. For example:

```
-> no ip multicast spoofing static-source-ip
-> no ip multicast vlan 2 spoofing static-source-ip
-> no ip multicast service 10 spoofing static-source-ip 10.2.2.1
```

Disabling the IGMP Spoofing

To disable IGMP spoofing on the system if no VLAN or SPB service is specified, use the **ip multicast spoofing** command as shown below:

```
-> ip multicast spoofing disable
```

Or, as an alternative, enter the following command to restore the IGMP spoofing to its default setting:

```
-> no ip multicast spoofing
```

You can also disable IGMP spoofing on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 spoofing disable
-> ip multicast service 10 spoofing disable
```

Or, as an alternative, enter the following commands to restore the IGMP spoofing to its default setting:

```
-> no ip multicast vlan 2 spoofing
-> no ip multicast service 10 spoofing
```

Enabling and Disabling the IGMP Zapping

By default, IGMP zapping (i.e., processing membership and source filter removals immediately without waiting for the specified time period for the protocol – this mode facilitates IP TV applications looking for quick changes between IP multicast groups) is disabled on a switch. The following subsections describe how to enable and disable IGMP zapping by using the **ip multicast zapping** command.

Enabling the IGMP Zapping

To enable IGMP zapping on the system if no VLAN or SPB service is specified, use the **ip multicast zapping** command as shown below:

```
-> ip multicast zapping enable
```

You can also enable IGMP zapping on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 zapping enable
-> ip multicast service 10 zapping enable
```

Disabling the IGMP Zapping

To disable IGMP zapping on the system if no VLAN or SPB service is specified, use the **ip multicast zapping** command as shown below:

```
-> ip multicast zapping disable
```

Or, as an alternative, enter the following command to restore the IGMP zapping to its default setting:

```
-> no ip multicast zapping
```

You can also disable IGMP zapping on the specified VLAN or SPB service by entering:

```
-> ip multicast vlan 2 zapping disable
-> ip multicast service 10 zapping disable
```

Or, as an alternative, enter the following commands to restore the IGMP zapping to its default setting:

```
-> no ip multicast vlan 2 zapping
-> no ip multicast service 10 zapping
```

Limiting IGMP Multicast Groups

By default there is no limit on the number of IGMP groups that can be learned on a port/VLAN/SPB service instance. A maximum group limit can be set on a port, VLAN, SPB service, or on a global level to limit the number of IGMP groups that can be learned. Once the configured limit is reached, a configurable action decides whether the new IGMP report is dropped or replaces an existing IGMP membership.

The maximum group limit can be applied globally, per VLAN, per SPB service, or per port. Port settings override VLAN and SPB service settings, which override global settings.

If the maximum number of groups is reached, an action can be configured to either drop the new membership request or replace an existing group membership as shown below.

Setting the IGMP Group Limit

To set the IGMP global group limit and drop any requests above the limit, use the **ip multicast max-group** command as shown below:

```
-> ip multicast max-group 25 action drop
```

To set the IGMP group limit for a VLAN or SPB service and replace an existing session, use the **ip multicast max-group** command as shown below:

```
-> ip multicast vlan 10 max-group 25 action replace
-> ip multicast service 10 max-group 25 action replace
```

To set the IGMP group limit for a port or an SPB Service Access Point (SAP), use the **ip multicast port max-group** command. For example, the following commands set the maximum group limit to 25 on port 1/1/13 and SAP port 1/1/23:10:

```
-> ip multicast port 1/1 max-group 25 action drop
-> ip multicast sap port 1/1/23:10 max-group 25 action drop
```

In the above example, 1/1/23:10 serves as a SAP ID in an SPB network. A SAP ID is comprised of an access port number (1/1/23) and an encapsulation value (10). Traffic received on access port 1/1/23 that is tagged with VLAN 10 is encapsulated and then forwarded on the SPB service that is associated with the SAP. Refer to Chapter 10, “Service Manager Commands,” for more information.

IPMSv6 Overview

An IPv6 multicast address identifies a group of nodes. A node can belong to any number of multicast groups. IPv6 multicast addresses are classified as fixed scope multicast addresses and variable scope multicast addresses.(See the "Reserved IPv6 Multicast Addresses" on page 27-29.)

IPMSv6 tracks the source VLAN or the source Shortest Path Bridging (SPB) service on which the Multicast Listener Discovery Protocol (MLD) requests are received. The network interfaces verify that a multicast packet is received by the switch on the source (or expected) port.

IPMSv6 Example

The figure on the following page shows an IPMSv6 network where video content can be provided to clients that request it. A server is attached to the switch that provides the source (i.e, multicast) IPv6 addresses. Clients from two different attached networks send MLD reports to the switch to receive the video content.

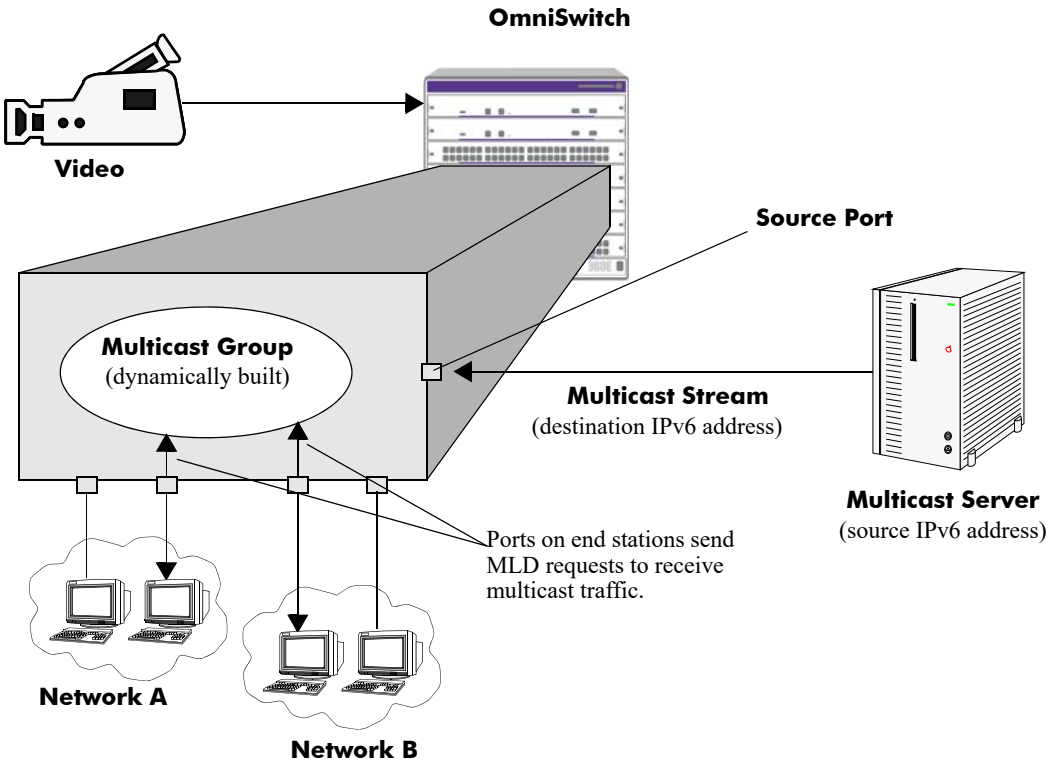


Figure 27-2 : IPMSv6 Example

Reserved IPv6 Multicast Addresses

The Internet Assigned Numbers Authority (IANA) classified the scope for IPv6 multicast addresses as fixed scope multicast addresses and variable scope multicast addresses. However, as the table below shows only well-known addresses, which are reserved and cannot be assigned to any multicast group.

Address	Description
FF00:0:0:0:0:0:0:0	reserved
FF01:0:0:0:0:0:0:0	node-local scope address
FF02:0:0:0:0:0:0:0	link-local scope
FF03:0:0:0:0:0:0:0	unassigned
FF04:0:0:0:0:0:0:0	unassigned
FF05:0:0:0:0:0:0:0	site-local scope
FF06:0:0:0:0:0:0:0	unassigned
FF07:0:0:0:0:0:0:0	unassigned
FF08:0:0:0:0:0:0:0	organization-local scope
FF09:0:0:0:0:0:0:0	unassigned
FF0A:0:0:0:0:0:0:0	unassigned
FF0B:0:0:0:0:0:0:0	unassigned
FF0C:0:0:0:0:0:0:0	unassigned
FF0D:0:0:0:0:0:0:0	unassigned
FF0E:0:0:0:0:0:0:0	global scope
FF0F:0:0:0:0:0:0:0	reserved

MLD Version 2

MLD is used by IPv6 systems (hosts and routers) to report their IPv6 multicast group memberships to any neighboring multicast routers. MLD Version 1 (MLDv1) handles forwarding by IPv6 multicast destination addresses only. MLD Version 2 (MLDv2) handles forwarding by source IPv6 addresses and IPv6 multicast destination addresses. Both MLDv1 and MLDv2 are supported.

Note. See "Configuring the MLD Version 2" on page 27-32 for information on configuring the IGMP version.

MLDv2 uses source filtering and reports multicast memberships to neighboring routers by sending membership reports. MLDv2 also supports Source Specific Multicast (SSM) by allowing hosts to report interest in receiving packets only from specific source addresses or from all but specific source addresses.

Configuring IPMSv6 on a Switch

This section describes how to use Command Line Interface (CLI) commands to complete the following configuration tasks:

- Enable and disable IPv6 Multicast Switching (IPMSv6) switch wide (see "Enabling and Disabling IPv6 Multicast Status" on page 27-30).
- Configure a port as an MLD static neighbor (see "Configuring and Removing an MLD Static Neighbor" on page 27-33).
- Configure a port as an MLD static querier (see "Configuring and Removing an MLD Static Querier" on page 27-34).
- Configure a port as an MLD static group (see "Configuring and Removing an MLD Static Group" on page 27-35)

Note. See the "IP Multicast Switching Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for complete documentation of IPMSv6 CLI commands.

Enabling and Disabling IPv6 Multicast Status

IPv6 Multicast is disabled by default on a switch. The following subsections describe how to enable and disable IPv6 Multicast by using the **ipv6 multicast admin-state** command.

Note. If IPv6 Multicast switching and routing is enabled on the system, the VLAN or SPB service configuration overrides the configuration of the system.

Enabling IPv6 Multicast Status

To enable IPv6 Multicast switching and routing on the system if no VLAN or SPB service is specified, use the **ipv6 multicast admin-state** command as shown below:

```
-> ipv6 multicast admin-state enable
```

You can also enable IPv6 Multicast switching and routing on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 admin-state enable
-> ipv6 multicast service 10 admin-state enable
```

Disabling IPv6 Multicast Status

To disable IPv6 Multicast switching and routing on the system if no VLAN or SPB service is specified, use the **ipv6 multicast admin-state** command as shown below:

```
-> ipv6 multicast admin-state disable
```

Or, as an alternative, enter the following command to restore the IPv6 Multicast status to its default setting:

```
-> no ipv6 multicast admin-state
```

You can also disable IPv6 Multicast on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 admin-state disable
-> ipv6 multicast service 10 admin-state disable
```

Or, as an alternative, enter the following commands to restore the IPv6 Multicast status to its default setting:

```
-> no ipv6 multicast vlan 2 admin-state
-> no ipv6 multicast service 10 admin-state
```

Enabling and Disabling Flooding of Unknown Multicast Traffic

When a traffic flow is first seen on a port. there is a brief period of time where traffic may get dropped before the forwarding information is calculated. When flooding unknown multicast traffic is enabled, no packets are dropped before the forwarding information is available.

By default, the flooding of unknown multicast traffic is disabled.The following subsections describe how to enable and disable this function by using the **ipv6 multicast flood-unknown** command.

Enabling Flooding of Unknown Multicast Traffic

You can enable the flooding of unknown multicast traffic on the system by entering **ipv6 multicast flood-unknown** followed by the **enable** keyword. For example:

```
-> ipv6 multicast flood-unknown enable
```

You can also enable the flooding of unknown multicast traffic on a specific VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 flood-unknown enable
-> ipv6 multicast service 10 flood-unknown enable
```

Disabling Flooding of Unknown Multicast Traffic

You can disable the flooding of unknown multicast traffic on the system by entering **ipv6 multicast flood-unknown** followed by the **disable** keyword. For example:

```
-> ipv6 multicast flood-unknown disable
```

Or, as an alternative, enter the following command to restore the flooding of unknown multicast traffic to its default setting:

```
-> no ipv6 multicast flood-unknown
```

You can also disable the flooding of unknown multicast traffic on the specified VLAN or SPB service by entering**:**

```
-> ipv6 multicast vlan 2 flood-unknown disable
-> ipv6 multicast service 10 flood-unknown disable
```

Or, as an alternative, enter the following commands to restore the flooding of unknown multicast traffic to its default setting:

```
-> no ipv6 multicast vlan 2 flood-unknown
-> no ipv6 multicast service 10 flood-unknown
```

Enabling and Disabling MLD Querier-forwarding

By default, MLD querier-forwarding is disabled.The following subsections describe how to enable and disable MLD querier-forwarding by using the **ipv6 multicast querier-forwarding** command.

Enabling the MLD Querier-forwarding

You can enable the MLD querier-forwarding by entering **ipv6 multicast querier-forwarding** followed by the **enable** keyword. For example, to enable the MLD querier-forwarding on the system if no VLAN or SPB service is specified, you would enter:

```
-> ipv6 multicast querier-forwarding enable
```

You can also enable the MLD querier-forwarding on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 querier-forwarding enable
-> ipv6 multicast service 10 querier-forwarding enable
```

Disabling the MLD Querier-forwarding

You can disable MLD querier-forwarding by entering **ipv6 multicast querier-forwarding** followed by the **disable** keyword. For example, to disable MLD querier-forwarding on the system if no VLAN or SPB service is specified, you would enter:

```
-> ipv6 multicast querier-forwarding disable
```

Or, as an alternative, enter the following command to restore MLD querier-forwarding to its default setting:

```
-> no ipv6 multicast querier-forwarding
```

You can also disable MLD querier-forwarding on the specified VLAN or SPB service by entering**:**

```
-> ipv6 multicast vlan 2 querier-forwarding disable
-> ipv6 multicast service 10 querier-forwarding disable
```

Or, as an alternative, enter the following commands to restore MLD querier-forwarding to its default setting:

```
-> no ipv6 multicast vlan 2 querier-forwarding
-> no ipv6 multicast service 10 querier-forwarding
```

Configuring and Restoring the MLD Version

By default, the version of Multicast Listener Discovery (MLD) Protocol is Version 1. The following subsections describe how to configure the MLD version as Version 1 or Version 2 by using the **ipv6 multicast version** command.

Configuring the MLD Version 2

To change the MLD version to Version 2 (MLDv2) on the system if no VLAN or SPB service is specified, use the **ipv6 multicast version** command as shown below:

```
-> ipv6 multicast version 2
```

Restoring the MLD Version 1

To restore the MLD version to Version 1 (MLDv1) on the system if no VLAN or SPB service is specified, use the **ipv6 multicast version** command by entering:

```
-> ipv6 multicast version 0
```

Or, as an alternative, enter the following command to restore the MLD version to Version 1:

```
-> no ipv6 multicast version
```

You can also restore the MLD version to Version 1 (MLDv1) on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 version 0
-> ipv6 multicast service 10 version 0
```

Or, as an alternative, enter the following commands to restore the MLD version to Version 1:

```
-> no ipv6 multicast vlan 2 version
-> no ipv6 multicast service 10 version
```

Configuring and Removing an MLD Static Neighbor

MLD static neighbor ports receive all multicast streams on the designated VLAN or SPB service and also receive MLD reports for the VLAN or SPB service. The following subsections describe how to configure and remove a static neighbor port by using the **ipv6 multicast static-neighbor** command.

Configuring an MLD Static Neighbor

To configure a port as an MLD static neighbor port, use the **ipv6 multicast static-neighbor** command with the **port** option. For example, the following command configures port 1/1/13 in VLAN 2 as an MLD static neighbor:

```
-> ipv6 multicast static-neighbor vlan 2 port 1/1/13
```

To configure an SPB Service Access Point (SAP) as an MLD static neighbor port, use the **ipv6 multicast static-neighbor** command with the **sap port** option. For example, the following command configures SAP port 1/1/23:10 that is bound to SPB service 10 as an MLD static neighbor:

```
-> ipv6 multicast static-neighbor service 10 sap port 1/1/23:10
```

In this example, 1/1/23:10 serves as a SAP ID, which is comprised of an access port number (1/1/23) and an encapsulation value (10). SPB service 10 is mapped to SAP ID 1/1/23:10. Traffic received on access port 1/1/23 that is tagged with VLAN 10 is encapsulated and then forwarded on service 10 through the SPB network. Refer to Chapter 10, “Service Manager Commands,” for more information.

To create an MLD static neighbor entry on a link aggregate, use the **linkagg** parameter. For example:

```
-> ipv6 multicast static-neighbor vlan 2 linkagg 7
-> ipv6 multicast static-neighbor service 10 sap linkagg 10:100
```

Removing an MLD Static Neighbor

To reset the port so that it is no longer an MLD static neighbor port, use the **no** form of the **ipv6 multicast static-neighbor** command with the **vlan** and **port** parameters. For example, the following command removes port 1/1/13 with designated VLAN 2 as an MLD static neighbor:

```
-> no ipv6 multicast static-neighbor vlan 2 port 1/1/13
```

To reset the SAP port so that it is no longer an MLD static neighbor port, use the **no** form of the **ipv6 multicast static-neighbor** command with the **service** and **sap port** parameters. For example, the following command removes SAP port 1/1/23 with designated service 10 as an MLD static neighbor:

```
-> no ipv6 multicast static-neighbor service 10 sap port 1/1/23
```

Configuring and Removing an MLD Static Querier

MLD static querier ports receive MLD reports generated on the designated VLAN or SPB service. Unlike MLD neighbor ports, they do not receive all multicast streams. The following subsections describe how to configure and remove a static querier by using the **ipv6 multicast static-querier** command.

Configuring an MLD Static Querier

To configure a port as an MLD static querier port, use the **ipv6 multicast static-querier** command with the **port** option. For example, the following command configures port 1/1/13 in VLAN 2 as an MLD static neighbor:

```
-> ipv6 multicast static-querier vlan 2 port 1/1/13
```

To configure an SPB Service Access Point (SAP) as an MLD static querier port, use the **ipv6 multicast static-querier** command with the **sap port** option. For example, the following command configures SAP port 1/1/23:10 that is bound to SPB service 10 as an MLD static querier:

```
-> ipv6 multicast static-querier service 10 sap port 1/1/23:10
```

In this example, 1/1/23:10 serves as a SAP ID, which is comprised of an access port number (1/1/23) and an encapsulation value (10). SPB service 10 is mapped to SAP ID 1/1/23:10. Traffic received on access port 1/1/23 that is tagged with VLAN 10 is encapsulated and then forwarded on service 10 through the SPB network. Refer to Chapter 10, “Service Manager Commands,” for more information.

To create an MLD static querier entry on a link aggregate, use the **linkagg** parameter. For example:

```
-> ipv6 multicast static-querier vlan 2 linkagg 7
-> ipv6 multicast static-querier service 10 sap linkagg 10:100
```

Removing an MLD Static Querier

To reset the port so that it is no longer an MLD static querier port, use the **no** form of the **ipv6 multicast static-querier** command. For example, the following command removes port 1/1/13 with designated VLAN 2 as an MLD static querier:

```
-> no ipv6 multicast static-querier vlan 2 port 1/1/13
```

To reset the SAP port so that it is no longer an MLD static querier port, use the **no** form of the **ipv6 multicast static-querier** command with the **service** and **sap port** parameters. For example, the following command removes SAP port 1/1/23 with designated service 10 as an MLD static querier:

```
-> no ipv6 multicast static-querier service 10 sap port 1/1/23
```

Configuring and Removing an MLD Static Group

MLD static group ports receive MLD reports generated on the specified IPv6 Multicast group address. The following subsections describe how to configure and remove an MLD static group by using the **ipv6 multicast static-group** command.

Configuring an MLD Static Group

To configure an MLD static group entry on a port, use the **ipv6 multicast static-group** command with the **port** option. For example, the following command configures an MLD static group entry with an IPv6 address of ff05::5 on port 1/1/3 in VLAN 3:

```
-> ipv6 multicast static-group ff05::5 vlan 3 port 1/1/13
```

To configure an MLD static group entry on an SPB Service Access Point (SAP), use the **ipv6 multicast static-group** command with the **sap port** option. For example, the following command configures an MLD static group entry with IPv6 address ff05::5 on SAP port 1/1/23:10 that is bound to SPB service 10:

```
-> ipv6 multicast static-group ff05::5 service 10 sap port 1/1/23:10
```

In this example, 1/1/23:10 serves as a SAP ID, which is comprised of an access port number (1/1/23) and an encapsulation value (10). SPB service 10 is mapped to SAP ID 1/1/23:10. Traffic received on access port 1/1/23 that is tagged with VLAN 10 is encapsulated and then forwarded on service 10 through the SPB network. Refer to Chapter 10, "Service Manager Commands," for more information.

To create an MLD static group entry on a link aggregate, use the **linkagg** parameter. For example:

```
-> ipv6 multicast static-group ff05::5 vlan 2 linkagg 7
-> ipv6 multicast static-group ff05::5 service 10 sap linkagg 10:100
```

Removing an MLD Static Group

To remove an MLD static group entry from a port, use the **no** form of the **ipv6 multicast static-group** command with the **vlan** and **port** parameters. For example, the following command removes the MLD static group entry with an IPv6 address of ff05::5 on port 1/1/13:

```
-> no ipv6 multicast static-group ff05::5 vlan 3 port 1/1/13
```

To remove an MLD static group entry from a SAP port, use the **no** form of the **ipv6 multicast static-group** command with the **service** and **sap port** parameters. For example, the following command removes the MLD static group entry with an IPv6 address of ff05::5 on SAP port 1/1/23:

```
-> no ipv6 multicast static-group ff05::5 service 10 sap port 1/1/23
```

In the above example, 1/1/23:10 serves as a SAP ID in an SPB network. A SAP ID is comprised of an access port number (1/1/23) and an encapsulation value (10). Traffic received on access port 1/1/23 that is tagged with VLAN 10 is encapsulated and then forwarded on the SPB service that is associated with the SAP. Refer to Chapter 10, "Service Manager Commands," for more information.

Modifying IPMSv6 Parameters

The table in "IPMSv6 Default Values" on page 27-3 lists default values for IPMSv6 parameters. The following sections describe how to use CLI commands to modify these parameters.

Modifying the MLD Query Interval

The default IPMSv6 query interval (i.e., the time between MLD queries) is 125 in seconds. The following subsections describe how to configure a user-specified query interval value and restore it by using the **ipv6 multicast query-interval** command.

Configuring the MLD Query Interval

You can modify the MLD query interval from 1 to 65535 seconds by entering **ipv6 multicast query-interval** followed by the new value. For example, to set the MLD query interval to 60 seconds on the system if no VLAN or SPB service is specified, you would enter:

```
-> ipv6 multicast query-interval 160
```

You can also modify the MLD query interval on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 query-interval 160
-> ipv6 multicast service 10 query-interval 160
```

Restoring the MLD Query Interval

To restore the MLD query interval to its default value on the system if no VLAN or SPB service is specified, use the **ipv6 multicast query-interval** command by entering:

```
-> ipv6 multicast query-interval 0
```

Or, as an alternative, enter the following command to restore the MLD query interval to its default value:

```
-> no ipv6 multicast query-interval
```

You can also restore the MLD query interval to its default value on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 query-interval 0
-> ipv6 multicast service 10 query-interval 0
```

Or, as an alternative, enter the following commands to restore the MLD query interval to its default value:

```
-> no ipv6 multicast vlan 2 query-interval
-> no ipv6 multicast service 10 query-interval
```

Modifying the MLD Last Member Query Interval

The default MLD last member query interval (i.e., the time period to reply to an MLD query message sent in response to a leave group message) is 1000 milliseconds. The following subsections describe how to configure the MLD last member query interval and restore it by using the **ipv6 multicast last-member-query-interval** command.

Configuring the MLD Last Member Query Interval

You can modify the MLD last member query interval from 1 to 65535 milliseconds by entering **ipv6 multicast last-member-query-interval** followed by the new value. For example, to set the MLD last member query interval to 600 milliseconds on the system if no VLAN or SPB service is specified, you would enter:

```
-> ipv6 multicast last-member-query-interval 2200
```

You can also modify the MLD last member query interval on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 last-member-query-interval 2200
-> ipv6 multicast service 10 last-member-query-interval 2200
```

Restoring the MLD Last Member Query Interval

To restore the MLD last member query interval to its default value on the system if no VLAN or SPB service is specified, use the **ipv6 multicast last-member-query-interval** command by entering:

```
-> ipv6 multicast last-member-query-interval 0
```

Or, as an alternative, enter the following command to restore the MLD last member query interval to its default value:

```
-> no ipv6 multicast last-member-query-interval
```

You can also restore the MLD last member query interval on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 last-member-query-interval 0
-> ipv6 multicast service 10 last-member-query-interval 0
```

Or, as an alternative, enter the following commands to restore the MLD last member query interval to its default value:

```
-> no ipv6 multicast vlan 2 last-member-query-interval
-> no ipv6 multicast service 10 last-member-query-interval
```

Modifying the MLD Query Response Interval

The default MLD query response interval (i.e., the time period to reply to an MLD query message) is 10000 milliseconds. The following subsections describe how to configure the MLD query response interval and restore it by using the **ipv6 multicast query-response-interval** command.

Configuring the MLD Query Response Interval

You can modify the MLD query response interval from 1 to 65535 milliseconds by entering **ipv6 multicast last-member-query-interval** followed by the new value. For example, to set the MLD query response interval to 6000 milliseconds you would enter:

```
-> ipv6 multicast query-response-interval 20000
```

You can also modify the MLD query response interval on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 query-response-interval 20000
-> ipv6 multicast service 10 query-response-interval 20000
```

Restoring the MLD Query Response Interval

To restore the MLD query response interval to its default value on the system if no VLAN or SPB service is specified, use the **ipv6 multicast query-response-interval** command by entering:

```
-> ipv6 multicast query-response-interval 0
```

Or, as an alternative, enter the following command to restore the MLD query response interval to its default value:

```
-> no ipv6 multicast query-response-interval
```

You can also restore the MLD query response interval on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 query-response-interval 0
-> ipv6 multicast service 10 query-response-interval 0
```

Or, as an alternative, enter the following command to restore the MLD query response interval to its default value:

```
-> no ipv6 multicast vlan 2 query-response-interval
-> no ipv6 multicast service 10 query-response-interval
```

Enabling and Disabling Zero-based MLD Query

By default, an all-zero source IPv6 address is used for MLD query packets when a non-querier is querying the membership of a port. The following subsections describe how to enable and disable using a zero-based MLD query by using the **ipv6 multicast zero-based-query** command.

Enabling Zero-based MLD Query

You can enable the use of a zero-based MLD query on the system by entering **ipv6 multicast zero-based-query** followed by the **enable** keyword. For example:

```
-> ipv6 multicast zero-based-query enable
```

Or, as an alternative, enter the following command to restore the zero-based MLD query to its default setting (enabled):

```
-> no ipv6 multicast zero-based-query
```

You can also enable the use of a zero-based MLD query on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 zero-based-query enable
-> ipv6 multicast service 10 zero-based-query enable
```

Or, as an alternative, enter the following commands to restore the zero-based MLD query to its default setting (enabled) for the specified VLAN or SPB service:

```
-> no ipv6 multicast vlan 2 zero-based-query
-> no ipv6 multicast service 10 zero-based-query
```

Disabling Zero-based MLD Query

You can disable the use of a zero-based MLD query on the system by entering **ipv6 multicast zero-based-query** followed by the **disable** keyword. For example:

```
-> ipv6 multicast zero-based-query disable
```

You can also disable the use of a zero-based MLD query on the specified VLAN or SPB service by entering**:**

```
-> ipv6 multicast vlan 2 zero-based-query disable
-> ipv6 multicast service 10 zero-based-query disable
```

Modifying the MLD Router Timeout

The default MLD router timeout (i.e., expiry time of IPv6 multicast routers) is 90 seconds. The following subsections describe how to configure a user-specified router timeout value and restore it by using the **ipv6 multicast router-timeout** command.

Configuring the MLD Router Timeout

You can modify the MLD router timeout from 1 to 65535 seconds by entering **ipv6 multicast router-timeout** followed by the new value. For example, to set the MLD router timeout to 360 seconds on the system if no VLAN or SPB service is specified, you would enter:

```
-> ipv6 multicast router-timeout 360
```

You can also modify the MLD router timeout on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 router-timeout 360
-> ipv6 multicast service 10 router-timeout 360
```

Restoring the MLD Router Timeout

To restore the MLD router timeout to its default value on the system if no VLAN or SPB service is specified, use the **ipv6 multicast router-timeout** command by entering:

```
-> ipv6 multicast router-timeout 0
```

Or, as an alternative, enter the following command to restore the MLD router timeout to its default value:

```
-> no ipv6 multicast router-timeout
```

You can also restore the MLD router timeout on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 router-timeout 0
-> ipv6 multicast service 10 router-timeout 0
```

Or, as an alternative, enter the following command to restore the MLD router timeout to its default value:

```
-> no ipv6 multicast vlan 2 router-timeout
-> no ipv6 multicast service 10 router-timeout
```

Modifying the Source Timeout

The default source timeout (i.e., expiry time of IPv6 multicast sources) is 30 seconds. The following subsections describe how to configure a user-specified source timeout value and restore it by using the **ipv6 multicast source-timeout** command.

Configuring the Source Timeout

You can modify the source timeout from 1 to 65535 seconds by entering **ipv6 multicast source-timeout** followed by the new value. For example, to set the source timeout to 360 seconds on the system if no VLAN or SPB service is specified, you would enter:

```
-> ipv6 multicast source-timeout 60
```

You can also modify the source timeout on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 source-timeout 60
-> ipv6 multicast service 10 source-timeout 60
```

Restoring the Source Timeout

To restore the source timeout to its default value on the system if no VLAN or SPB service is specified, use the **ipv6 multicast source-timeout** command by entering:

```
-> ipv6 multicast source-timeout 0
```

Or, as an alternative, enter the following command to restore the source timeout to its default value:

```
-> no ipv6 multicast source-timeout
```

You can also restore the source timeout on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 source-timeout 0
-> ipv6 multicast service 10 source-timeout 0
```

Or, as an alternative, enter the following command to restore the source timeout to its default value:

```
-> no ipv6 multicast vlan 2 source-timeout
-> no ipv6 multicast service 10 source-timeout
```

Enabling and Disabling the MLD Querying

By default MLD querying is disabled.The following subsections describe how to enable and disable MLD querying by using the **ipv6 multicast querying** command.

Enabling the MLD Querying

You can enable the MLD querying by entering **ipv6 multicast querying** followed by the **enable** keyword. For example, to enable the MLD querying you would enter:

```
-> ipv6 multicast querying enable
```

You can also enable the MLD querying on the specified VLAN by entering**:**

```
-> ipv6 multicast vlan 2 querying enable
```

Specifying a Static Source IPv6 Address

By default, a source IPv6 address is not specified when MLD querying is enabled; the system automatically determines the addresses to use for MLD queries. However, a static source IPv6 address can be specified to overcome the need for an IPv6 interface. If configured, the static source IPv6 address is then always used for querying, regardless of the IPv6 interface address or administrative state.

To configure a static source IPv6 address for the system, use the **ipv6 multicast querying** command with the **static-source-ip** parameter. For example:

```
-> ipv6 multicast querying static-source-ip 10.2.2.1
```

To configure a static source IPv6 address for a specific VLAN, use the **ipv6 multicast querying** command with the **vlan** and **static-source-ip** parameters. For example:

```
-> ipv6 multicast querying vlan 2 static-source-ip 10.2.2.1
```

To remove a static source IPv6 address for the system or a specific VLAN, use the **no** form of the **ipv6 multicast querying** command. For example:

```
-> no ipv6 multicast querying static-source-ip
-> no ipv6 multicast querying vlan 2 static-source-ip
```

Disabling the MLD Querying

You can disable the MLD querying by entering **ipv6 multicast querying** followed by the **disable** keyword. For example, to disable the MLD querying you would enter:

```
-> ipv6 multicast querying disable
```

Or, as an alternative, enter the following command to restore the MLD querying to its default setting:

```
-> no ipv6 multicast querying
```

You can also disable the MLD querying on the specified VLAN by entering**:**

```
-> ipv6 multicast vlan 2 querying disable
```

Or, as an alternative, enter the following command to restore the MLD querying to its default setting:

```
-> no ipv6 multicast vlan 2 querying
```

Modifying the MLD Robustness Variable

The default value of the robustness variable (i.e., the variable that allows fine-tuning on the network, where the expected packet loss is greater) is 2. The following subsections describe how to set the value of the MLD robustness variable and restore it by using the **ipv6 multicast robustness** command.

Configuring the MLD Robustness Variable

You can modify the MLD robustness variable from 1 to 7 on the system if no VLAN or SPB service is specified, by entering **ipv6 multicast robustness**, followed by the new value. For example, to set the value of robustness to 3 you would enter:

```
-> ipv6 multicast robustness 3
```

Note. If the links are known to be lossy, then robustness can be set to a higher value (7).

You can also modify the MLD robustness variable from 1 to 7 on the specified VLAN or SPB service by entering**:**

```
-> ipv6 multicast vlan 2 robustness 3
-> ipv6 multicast service 10 robustness 3
```

Restoring the MLD Robustness Variable

You can restore the MLD robustness variable to its default value on the system if no VLAN or SPB service is specified by entering **ipv6 multicast robustness** followed by the value 0, as shown below:

```
-> ipv6 multicast robustness 0
```

Or, as an alternative, enter the following command to restore the MLD robustness to its default value:

```
-> no ipv6 multicast robustness
```

You can also modify the MLD robustness variable from 1 to 7 on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 robustness 0
-> ipv6 multicast service 10 robustness 0
```

Or, as an alternative, enter the following commands to restore the MLD robustness to its default value:

```
-> no ipv6 multicast vlan 2 robustness
-> no ipv6 multicast service 10 robustness
```

Enabling and Disabling MLD Spoofing

By default, MLD spoofing (i.e., replacing a client's MAC and IPv6 address with the system's MAC and IPv6 address, when proxying aggregated MLD group membership information) is disabled on the switch. The following subsections describe how to enable and disable spoofing by using the **ipv6 multicast spoofing** command.

Enabling MLD Spoofing

To enable MLD spoofing on the system if no VLAN or SPB service is specified, you use the **ipv6 multicast spoofing** command as shown below:

```
-> ipv6 multicast spoofing enable
```

You can also enable MLD spoofing on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 spoofing enable
-> ipv6 multicast service 10 spoofing enable
```

Specifying a Static Source IPv6 Address

By default, a source IPv6 address is not specified when MLD spoofing is enabled; the system automatically determines the addresses to use for spoofing. However, a static source IPv6 address can be specified to overcome the need for an IPv6 interface. If configured, the static source IPv6 address is then always used for spoofing, regardless of the IPv6 interface address or administrative state.

To configure a static source IPv6 address for the system, use the **ipv6 multicast spoofing** command with the **static-source-ip** parameter. For example:

```
-> ipv6 multicast spoofing static-source-ip 3333::1
```

To configure a static source IPv6 address for a specific VLAN or SPB service, use the **ipv6 multicast spoofing** command with the **vlan** and **static-source-ip** parameters. For example:

```
-> ipv6 multicast vlan 2 spoofing static-source-ip 3333::1
-> ipv6 multicast service 10 spoofing static-source-ip 333::1
```

To remove a static source IPv6 address for the system or a specific VLAN or SPB service, use the **no** form of the **ipv6 multicast spoofing** command. For example:

```
-> no ipv6 multicast spoofing static-source-ip
-> no ipv6 multicast vlan 2 spoofing static-source-ip
-> no ipv6 multicast service 10 spoofing static-source-ip
```

Disabling MLD Spoofing

To disable MLD spoofing on the system if no VLAN or SPB service is specified, you use the **ipv6 multicast spoofing** command as shown below:

```
-> ipv6 multicast spoofing disable
```

Or, as an alternative, enter the following command to restore the MLD spoofing to its default setting:

```
-> no ipv6 multicast spoofing
```

You can also disable MLD spoofing on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 spoofing disable
-> ipv6 multicast service 10 spoofing disable
```

Or, as an alternative, enter the following command to restore the MLD spoofing to its default setting:

```
-> no ipv6 multicast vlan 2 spoofing
-> no ipv6 multicast service 10 spoofing
```

Enabling and Disabling the MLD Zapping

By default MLD (i.e., processing membership and source filter removals immediately without waiting for the specified time period for the protocol– this mode facilitates IP TV applications looking for quick changes between IP multicast groups.) is disabled on a switch. The following subsections describe how to enable and disable zapping by using the **ipv6 multicast zapping** command.

Enabling the MLD Zapping

To enable MLD zapping on the system if no VLAN or SPB service is specified, use the **ipv6 multicast zapping** command as shown below:

```
-> ipv6 multicast zapping enable
```

You can also enable MLD zapping on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 zapping enable
-> ipv6 multicast service 10 zapping enable
```

Disabling the MLD Zapping

To disable MLD zapping on the system if no VLAN or SPB service is specified, use the **ipv6 multicast zapping** command as shown below:

```
-> ipv6 multicast zapping disable
```

Or, as an alternative, enter the following command to restore MLD zapping to its default setting:

```
-> no ipv6 multicast zapping
```

You can also disable MLD zapping on the specified VLAN or SPB service by entering:

```
-> ipv6 multicast vlan 2 zapping disable
-> ipv6 multicast service 10 zapping disable
```

Or, as an alternative, enter the following commands to restore MLD zapping to its default setting:

```
-> no ipv6 multicast vlan 2 zapping
-> no ipv6 multicast service 10 zapping
```

Limiting MLD Multicast Groups

By default there is no limit on the number of MLD groups that can be learned on a port/VLAN/SPB service instance. A maximum group limit can be set on a port, VLAN, SPB service, or on a global level to limit the number of MLD groups that can be learned. Once the configured limit is reached, a configurable action decides whether the new MLD report is dropped or replaced an existing MLD membership.

The maximum group limit can be applied globally, per VLAN, per SPB service, or per port. Port settings override VLAN and SPB service settings, which override global settings.

If the maximum number of groups is reached an action can be configured to either drop the new membership request or replace an existing group membership as show below.

Setting the MLD Group Limit

To set the MLD global group limit and drop any requests above the limit, use the **ipv6 multicast max-group** command as shown below:

```
-> ipv6 multicast max-group 25 action drop
```

To set the MLD group limit for a VLAN or SPB service and replace any requests above the limit, use the **ipv6 multicast max-group** command as shown below:

```
-> ipv6 multicast vlan 2 max-group 25 action replace
-> ipv6 multicast service 10 max-group 25 action replase
```

To set the MLD group limit for a port and drop any requests above the limit, use the **ipv6 multicast port max-group** command as shown below:

```
-> ipv6 multicast port 1/1 max-group 25 action drop
```

In the above example, 1/1/23:10 serves as a SAP ID in an SPB network. A SAP ID is comprised of an access port number (1/1/23) and an encapsulation value (10). Traffic received on access port 1/1/23 that is tagged with VLAN 10 is encapsulated and then forwarded on the SPB service that is associated with the SAP. Refer to Chapter 10, “Service Manager Commands,” for more information.

IPMS Application Example

The figure below shows a sample network with the switch sending multicast video. A client attached to Port 5 needs to be configured as a static IGMP neighbor and another client attached to Port 2 needs to be configured as a static IGMP querier.

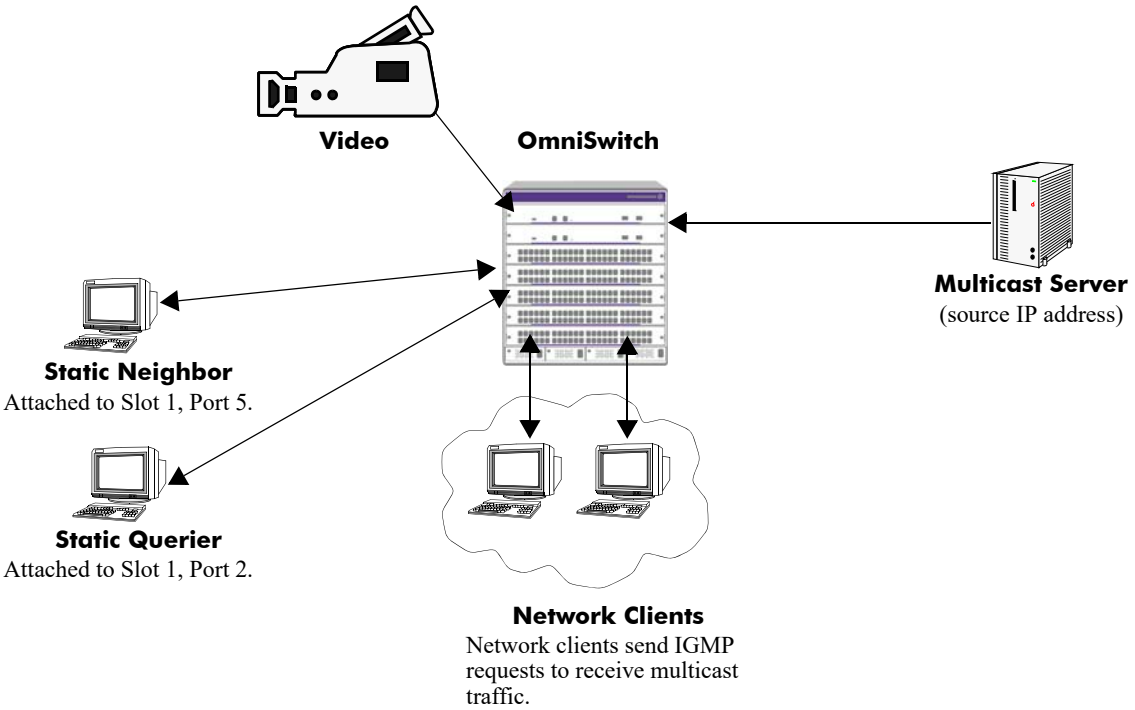


Figure 27-3 : Example IPMS Network

The network administrator has determined that the network is too lossy and therefore the robustness variable needs to be set to a higher (i.e., 7) value.

Follow the steps below to configure this network:

Note. All the steps following Step 1 (which must be executed first) can be entered in any order.

1. Enable IP Multicast Switching and Routing switch-wide, by entering:

```
-> ip multicast admin-state enable
```

2. Configure the client attached to Port 5 as a static neighbor belonging to VLAN 5 by entering:

```
-> ip multicast static-neighbor vlan 5 port 1/5
```

3. Configure the client attached to Port 2 as a static querier belonging to VLAN 5 by entering:

```
-> ip multicast static-querier vlan 5 port 1/2
```

4. Modify the robustness variable from its default value of 2 to 7 by entering:

```
-> ip multicast robustness 7
```

An example of what these commands look like entered sequentially on the command line:

```
-> ip multicast admin-state enable
-> ip multicast static-neighbor vlan 5 port 1/5
-> ip multicast static-querier vlan 5 port 1/2
-> ip multicast robustness 7
```

As an option, you can use the **show ip multicast**, **show ip multicast neighbor**, and **show ip multicast querier** commands to confirm your settings as shown below:

```
-> show ip multicast

Status:                                          = Enabled
Querying:                                        = Disabled
Proxying:                                        = Disabled
Spoofing:                                        = Disabled
Zapping:                                         = Disabled
Querier Forwarding:                                = Disabled
Version:                                         = 1
Robustness:                                      = 2
Query Interval (seconds):                        = 125
Query Response Interval (milliseconds):          = 10000
Last Member Query Interval(milliseconds):        = 1000
Unsolicited Report Interval (seconds)            = 1,
Router Timeout (seconds):                        = 90
Source Timeout (seconds):                        = 30

-> show ip multicast neighbor

Total 1 Neighbors
Host Address    VLAN  Port  Static  Count  Life
---------------+-----+-----+-------+------+-----
1.0.0.2          5     1/5   no      1      86


-> show ip multicast querier

Total 1 Queriers
Host Address    VLAN  Port  Static  Count  Life
---------------+-----+-----+-------+------+-----
1.0.0.3          5     1/2   no      1      250
```

IPMSv6 Application Example

The figure below shows a sample network with the switch sending multicast video. A client attached to Port 5 needs to be configured as a static MLD neighbor and another client attached to Port 2 needs to be configured as a static MLD querier.

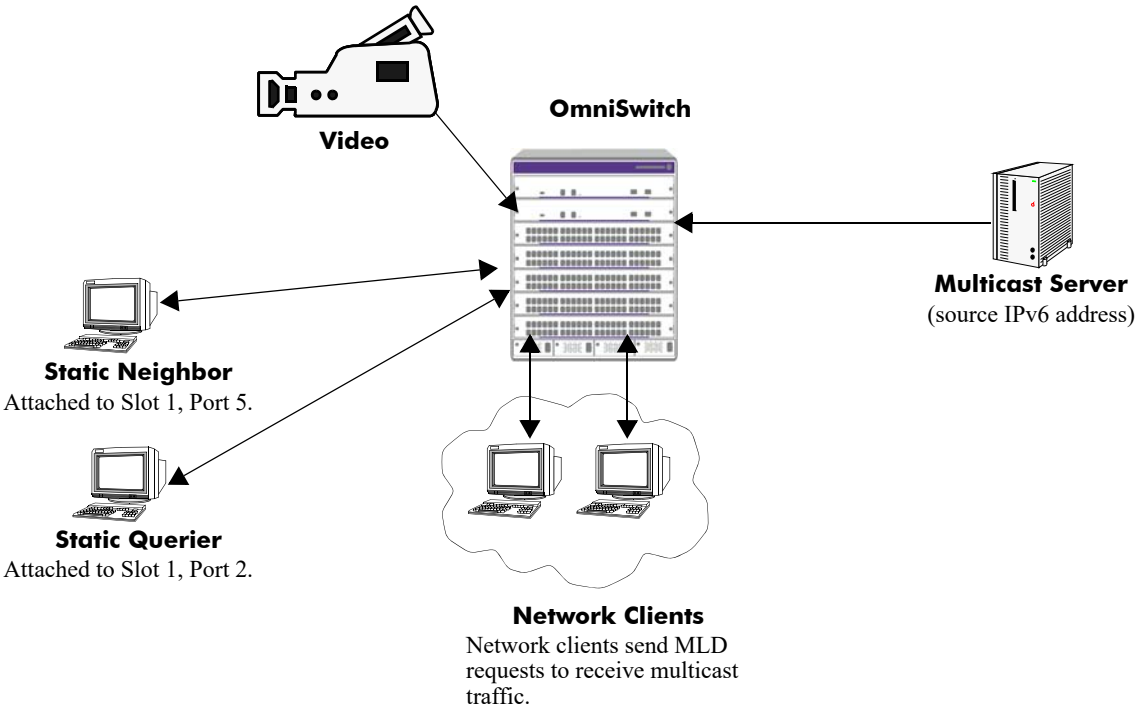


Figure 27-4 : Example IPMS v6 Network

The network administrator has determined that the network is too lossy and therefore the robustness variable needs to be set to a higher (i.e., 7) value.

Follow the steps below to configure this network:

Note. All the steps following Step 1 (which must be executed first) can be entered in any order.

1. Enable IP Multicast Switching and Routing switch-wide, by entering:

```
-> ipv6 multicast admin-state enable
```

2. Configure the client attached to Port 5 as a static MLD neighbor belonging to VLAN 5 by entering:

```
-> ipv6 multicast static-neighbor vlan 5 port 1/5
```

3. Configure the client attached to Port 2 as a static MLD querier belonging to VLAN 5 by entering:

```
-> ipv6 multicast static-querier vlan 5 port 1/2
```

4. Modify the robustness variable from its default value of 2 to 7 by entering:

```
-> ipv6 multicast robustness 7
```

An example of what these commands look like entered sequentially on the command line:

```
-> ipv6 multicast admin-state enable
-> ipv6 multicast static-neighbor vlan 5 port 1/5
-> ipv6 multicast static-querier vlan 5 port 1/2
-> ipv6 multicast robustness 7
```

As an option, you can use the **show ipv6 multicast**, **show ipv6 multicast neighbor**, and **show ipv6 multicast querier** commands to confirm your settings as shown below:

```
-> show ipv6 multicast

Status:                                          = Enabled
Querying:                                        = Disabled
Proxying:                                        = Disabled
Spoofing:                                        = Disabled
Zapping:                                         = Disabled
Querier Forwarding:                              = Disabled
Version:                                         = 1
Robustness:                                      = 2
Query Interval (seconds):                        = 125
Query Response Interval (milliseconds):          = 10000
Last Member Query Interval(milliseconds):        = 1000
Unsolicited Report Interval (seconds)            = 1,
Router Timeout (seconds):                        = 90
Source Timeout (seconds):                        = 30

-> show ipv6 multicast neighbor

Total 1 Neighbors
Host Address              VLAN  Port  Static  Count  Life
--------------------------+-----+-----+-------+------+-----
fe80::2a0:ccff:fed3:2853   5      1/5   no      1      6

-> show ipv6 multicast querier

Total 1 Queriers
Host Address              VLAN  Port  Static  Count  Life
--------------------------+-----+-----+-------+------+-----
fe80::2a0:ccff:fed3:2854   5      1/2   no      1      6
```

Displaying IPMS Configurations and Statistics

The OmniSwitch IP Multicast Switching (IPMS) **show** commands provide tools to monitor IPMS traffic and settings and to troubleshoot problems. These commands are described below:

show ip multicast	Displays the general IP Multicast switching and routing configuration parameters on a switch.
show ip multicast group	Displays all detected multicast groups that have members. If you do not specify an IP address then all multicast groups on the switch is displayed.
show ip multicast neighbor	Displays all neighboring multicast routers.
show ip multicast querier	Displays all multicast queriers.
show ip multicast port	Displays the IPMS multicast forwarding table. If you do not specify a multicast group IP address, then the forwarding table for all multicast groups are displayed.
show ip multicast source **show ip multicast forward** **show ip multicast bridge** **show ip multicast bridge-forward**	Displays the IPMS multicast source/forwarding table. If you do not specify a multicast group IP address, then the tables for all multicast groups are displayed.
show ip multicast tunnel	Displays the IP multicast switch and routing tunneling table entries matching the specified IP multicast group address, or all the entries if no IP multicast address is specified.

To get a quick look at the IPMS groups on your switch, use the **show ip multicast group** command. For example:

```
-> show ip multicast group domain vlan

Total 3 Groups

Group Address   Source Address  VLAN  Port  Mode     Static  Count  Life
---------------+---------------+-----+-----+--------+-------+------+-----
231.0.0.3       1.0.0.5         1     2/1   exclude  no      1      257
234.0.0.4       0.0.0.0         1     2/1   exclude  no      1      218
229.0.0.1       0.0.0.0         1     2/13  exclude  yes     0      0

-> show ip multicast group domain service

Total 3 Groups

Group Address   Source Address  Service  Interface     Mode     Static  Count  Life
---------------+---------------+--------+-------------+--------+-------+------+----
225.51.1.1      0.0.0.0          1001    sdp:32778     exclude  no      224    189
225.51.1.1      0.0.0.0          1001    sdp:32779     exclude  no      225    241
225.51.1.1      0.0.0.0          1001    sdp:32822     exclude  no      225    242
```

Note. See the "IP Multicast Switching Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for complete documentation on IPMS **show** commands.

Displaying IPMSv6 Configurations and Statistics

The OmniSwitch IPv6 Multicast Switching (IPMSv6) **show** commands provide tools to monitor IPMSv6 traffic and settings and to troubleshoot problems. These commands are described below:

show ipv6 multicast	Displays the general IPv6 Multicast switching and routing configuration parameters on a switch.
show ipv6 multicast group	Displays all detected multicast groups that have members. If you do not specify an IPv6 address, then all multicast groups on the switch are displayed.
show ipv6 multicast neighbor	Displays all neighboring IPv6 multicast routers.
show ipv6 multicast querier	Displays all IPv6 multicast queriers.
show ipv6 multicast port	Displays the IPMSv6 multicast forwarding table. If you do not specify a multicast group IPv6 address, then the forwarding table for all multicast groups are displayed.
show ipv6 multicast source **show ipv6 multicast forward** **show ipv6 multicast bridge** **show ipv6 multicast bridge-forward**	Displays the IPMSv6 multicast source/forwarding table. If you do not specify a multicast group IPv6 address, then the table for all multicast groups are displayed.
show ipv6 multicast tunnel	Display the IPv6 multicast switch and routing tunneling table entries matching the specified IPv6 multicast group address, or all the entries if no IPv6 multicast address is specified.

To get a quick look at the IPMSv6 groups on the switch, use the **show ipv6 multicast group** command. For example:

```
-> show ipv6 multicast group domain vlan

Total 3 Groups

Group Address    Source Address  VLAN  Port  Mode     Static  Count  Life
----------------+---------------+-----+-----+--------+-------+------+-----
ff05::5          ::               1     2/1   exclude  no      1      145
ff05::6          3333::1          1     2/1   exclude  no      1      242
ff05::9          ::               1     2/13  exclude  yes     0      0

-> show ipv6 multicast group domain service

Total 3 Groups

Group Address  Source Address  Service   Interface   Mode   Static   Count    Life
-------------+---------------+--------+-----------+--------+--------+------+------
ff05::5            ::          1       SAP:1/5:10  exclude  no       5520   172
ff05::6            ::          1       SAP:1/5:20  exclude  no       5520   172
ff05::7            ::          1       sdp:32776:1 exclude  no       5520   172
```

Note. See the "IP Multicast Switching Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for complete documentation on IPMS **show** commands.

Note. The OmniSwitch 6465/6560/9900 contain separate bridge and routing engines, the multicast software has to maintain the state of both of them to ensure proper multicast functionality. Due to the architectural differences the '**show ip multicast source**' and '**show ip multicast forward**' commands behave differently. These commands are only valid when multicast routing is enabled (OS9900 only). Sources on all interfaces that are enabled for multicast routing will be displayed along with the forward state for the RPF interface as determined by multicast routing.

- **show ip[v6] multicast forward** –Will display the forwarding state for the RPF interface as determined by multicast routing.
- **show ip[v6] multicast source** – Will display the sources on all interfaces that are enabled for multicast routing.

To provide similar capability as other platforms the '**show ip[v6] multicast bridge**' and '**show ip[v6] multicast bridge-forward**' commands can be used to display the forwarding database on an OmniSwitch 6465/6560/9900.

28 Configuring QoS

The OmniSwitch software and queue management architecture provide a way to identify traffic entering the network and manipulate flows coming through the switch. The flow manipulation (generally referred to as *Quality of Service* or *QoS*) can be as simple as configuring QoS policies to allow/deny traffic or as complicated as remapping 802.1p bits from a Layer 2 network to ToS values in a Layer 3 network.

The types of policies typically used include, but are not limited to, the following:

- **Basic QoS**—includes traffic prioritizing and bandwidth shaping.
- **ICMP policies**—includes filtering, prioritizing, and/or rate limiting ICMP traffic for security.
- **802.1p/ToS/DSCP**—includes policies for marking and mapping.
- **Policy Based Routing (PBR)**—includes policies for redirecting routed traffic.
- **Policy Based Mirroring**— includes mirror-to-port (MTP) policies for mirroring ingress, egress, or both ingress and egress traffic, and policy based multiple destination mirroring.
- **Access Control Lists (ACLs)**—ACLs are a specific type of QoS policy that is used for Layer 2 and Layer 3/4 filtering. See "Using Access Control Lists" on page 28-64.

This implementation of QoS integrates traffic management with QoS scheduling. Embedded profiles apply the QoS admission control and bandwidth management configurations to traffic flows.

Data Center Bridging (DCB) protocols are also supported and implemented using embedded profiles in the same manner that QoS profiles are applied. DCB and QoS profiles are mutually exclusive in that if the OmniSwitch Data Center software license is installed, only DCB profiles are applied. For more information, see the "Configuring Data Center Bridging" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

In This Chapter

This chapter describes QoS in general and how policies, port-based QoS configuration, and queue management are used on the switch. It provides information about configuring QoS through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following topics and procedures are included in this chapter:

- "QoS General Overview" on page 28-3.
- "Classification" on page 28-5.
- "Congestion Management" on page 28-11.
- "OmniSwitch Congestion Avoidance" on page 28-20.
- "Traffic Policing and Shaping" on page 28-21.
- "QoS Defaults" on page 28-34.
- "Configuring QoS" on page 28-38.
- "Policy Applications" on page 28-75.
- "Enhanced Congestion Notification (ECN)" on page 28-84

QoS General Overview

Quality of Service (QoS) refers to transmission quality and available service that is measured and sometimes guaranteed in advance for a particular type of traffic in a network. QoS lends itself to circuit-switched networks like ATM, which bundle traffic into cells of the same length and transmit the traffic over predefined virtual paths. In contrast, IP and other packet-switched networks operate on the concept of shared resources and *best effort* routing, using bandwidth as needed and reassembling packets at their destinations. Applying QoS to packet-switched networks requires different mechanisms than those used in circuit-switched networks.

QoS is often defined as a way to manage bandwidth. Another way to handle different types of flows and increased bandwidth requirements is to add more bandwidth. But bandwidth can be expensive, particularly at the WAN connection. If LAN links that connect to the WAN are not given more bandwidth, bottlenecks can still occur. Also, adding enough bandwidth to compensate for peak load periods mean that at times some bandwidth is unused. In addition, adding bandwidth does not guarantee any kind of control over network resources.

Using QoS, a network administrator can gain more control over networks where different types of traffic (or flows) are in use or where network congestion is high. Preferential treatment can be given to individual flows as required. Voice over IP (VoIP) traffic or mission-critical data can be marked as priority traffic and/or given more bandwidth on the link. QoS can also prevent large flows, such as a video stream, from consuming all the bandwidth on a link. Using QoS, a network administrator can decide which traffic needs preferential treatment and which traffic can be adequately served with best effort.

QoS is implemented on the switch through the use of user-defined policies, port-based QoS configuration, and integration with virtual output queuing to manage egress congestion. The following simplified illustration shows an example of how video traffic can receive priority over email traffic.

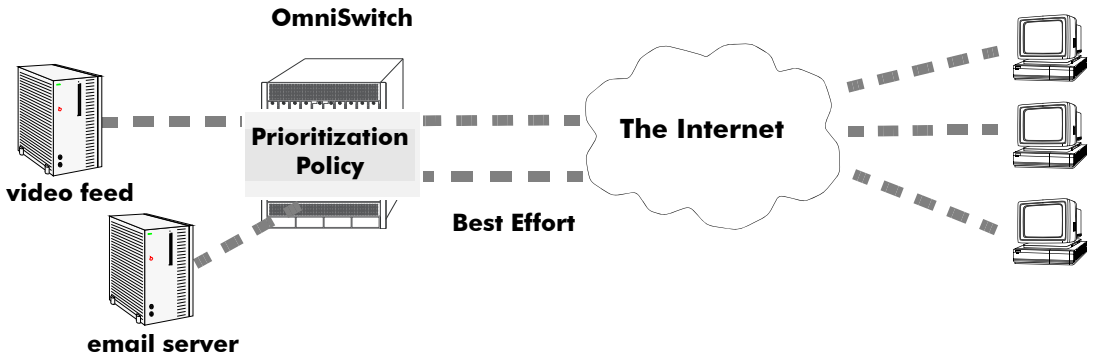


Figure 28-1 : Sample QoS Setup

QoS sits in the ingress and egress software path. IP calls QoS to validate packets destined for the switch. IP also calls QoS to validate and/or prioritize packets originating from the switch.

The general order of events with respect to the OmniSwitch implementation of QoS are as follows:

1 Classification—Packets are classified and marked according to policies and traffic behavior. This is accomplished on the ingress using technologies such as 802.1p, IP precedence and Diffserv Code Point (DSCP). See "Classification" on page 28-5 for more information.

2 Congestion Management—Classified packets are prioritized and placed into queues based on Class of Service (CoS) markings to ensure preferential treatment to high priority traffic. See "Congestion Management" on page 28-11.

3 Congestion Avoidance—Packets that are not high priority are randomly dropped to help avoid "tail drop" on the queues. See "OmniSwitch Congestion Avoidance" on page 28-20.

4 Traffic Policing and Shaping—Packet flows are policed or shaped to limit the rate of traffic received or sent by the switch. See "Traffic Policing and Shaping" on page 28-21.

Classification

Classification is the process of identifying certain types of network traffic (flows) and then, if necessary, marking a specific flow or group of flows with a priority (class of service) value. The class of service (CoS) value assigned is then used by other QoS features to determine how the flow is treated throughout the network.

The CoS value assigned to a specific flow is based on one of the following technologies:

- **IP Precedence**—Type of Service (ToS) or Differentiated Services (DiffServ).

  ToS refers to using the three precedence bits of the ToS field in an IP packet to specify a priority value ranging from 0 (lowest) to 7 (highest).

  DiffServ uses the DiffServ Code Point (DSCP) value specified in the first 6 bits of the ToS field. The DSCP determines the CoS by specifying a per-hop behavior (PHB) for a specific flow or group of flows. The PHB indicates the forwarding behavior of a flow by specifying bandwidth, queueing schemes, and criteria for dropping packets.

- **Layer 2 802.1p Priority**

  The 802.1p priority value is specified in the Tag Control Info (TCI) field of an Ethernet frame. This value also ranges from 0 (lowest) to 7 (highest) and maps to the ToS precedence values.

The OmniSwitch output queuing capability uses these CoS values to determine the forwarding treatment by prioritizing flows based on application and network requirements. For more information about output queue (congestion) management, see "Congestion Management" on page 28-11.

How Traffic is Classified and Marked

The OmniSwitch provides the following tools and techniques for classifying network traffic:

- **QoS Policy Rules**

  A policy (or a *policy rule*) is made up of a condition and an action. The condition specifies parameters that the switch examines in incoming flows, such as destination address or Type of Service (ToS) bits. The action specifies what the switch does with a flow that matches the condition; for example, it can queue the flow with a higher priority, or reset the ToS bits. See "QoS Policy Overview" on page 28-29 for more information.

- **Access Control Lists (ACLs)**

  ACLs are QoS policies used to control whether or not packets are allowed or denied at the switch or router interface. ACLs are sometimes referred to as filtering lists and may also specify priority-setting actions. See "Using Access Control Lists" on page 28-64 for more information.

- **Port-based QoS**

  Individual ports are configured to either trust (recognize) or not trust (do not recognize) existing 802.1p or ToS/DSCP values within a packet or to apply a user-defined default classification value. Port-based QoS often works in conjunction with QoS policy rules to prioritize packet flows. By default, all switch ports are untrusted. See "Configuring Trusted Ports" on page 28-9 for more information.

When packets ingress on a switch port, the packets are classified and marked as follows:

- If a packet matches a QoS policy rule that sets a new priority value (802.1p or ToS/DSCP), the egress priority for the packet is set using the value specified in the rule.

- If a packet ingresses on a *trusted* port and does not match any QoS policy that sets priority, then the existing 802.1p value (non-IP packets) or the ToS/DSCP value (IP packets) or the default classification priority configured for the port is used to determine priority for the packet.
- If a packet ingresses on an *untrusted* port and does not match any QoS policy that sets priority, then the default 802.1p value configured for the port (tagged/untagged non-IP packets) or the default ToS/DSCP value configured for the port (IP packets) is used to determine priority for the packet.
- If the default classification value for the port is set to DSCP, the DSCP value of a tagged IP packet is mapped to the 802.1p value for that same packet. In other words, the 802.1p priority is overwritten with the precedence bits of the DSCP value. This does not apply to Layer 2 packets. See "Maintaining the 802.1p Priority for IP Packets" on page 28-65 for more information.
- The egress priority for a packet ingressing on a VLAN Stacking port (a trusted port) is set using the existing 802.1p value or configured through an associated VLAN Stacking service.
- IP phone traffic is automatically trusted by default. See "Automatic QoS Prioritization for IP Phone Traffic" on page 28-6 for more information.

Classifying Bridged Traffic as Layer 3

In some network configurations it is required to force the switch to classify bridged traffic as routed (Layer 3) traffic. Typically this option is used for QoS filtering. See "Using Access Control Lists" on page 28-64 for more information about filtering.

The Layer 3 classification of bridged traffic is no different from the classification of normal Layer 3 routed traffic. Note that this implementation of QoS always performs Layer 3 classification of bridged traffic; it is not an option. As a result,

- Layer 3 ACLs are always effected on bridged traffic.
- The switch can bridge and route traffic to the same destination.
- All IP packets are prioritized based on ToS if the default classification on the port is set to DSCP. If DSCP is not the default classification, then the IP packets are prioritized based on 802.1p.

Note that Layer 3 ACLs are effected on bridged IP traffic and Layer 2 ACLs are effected on routed traffic.

Automatic QoS Prioritization for IP Phone Traffic

Automatic QoS prioritization refers to prioritizing certain subsets of switch traffic without having to configure a specific QoS policy to do the same for each type of traffic. This functionality is currently available for IP phone traffic.

The switch automatically trusts the priority of IP phone traffic by default. This means that the priority value contained in packets originating from IP phones is used for the ingress priority. The default priority value configured for the QoS port receiving such traffic is used for the egress priority of the packet.

IP phone traffic is identified by examining the source MAC address of the packet received on the port. If the source MAC falls within one of the following ranges, the QoS IP phone priority is automatically assigned to the MAC:

MAC Address Range	Description
00:80:9f:00:00:00—00:80:9f:ff:ff:ff	Enterprise IP Phones Range
78:81:02:00:00:00—78:81:02:ff:ff:ff	Communications IP Phones Range
00:13:fa:00:00:00—00:13:fa:ff:ff:ff	Lifesize IP Phones Range
48:7a:55:00:00:00—48:7a:55:ff:ff:ff	ALE 8008 IP Phone MAC Range

In addition to prioritizing IP phone traffic, it is also possible to automatically prioritize non-IP phone traffic. This is done by adding up to four MAC addresses or four ranges of MAC addresses to the predefined QoS "alaPhones" MAC address group. See "Creating MAC Groups" on page 28-58 for more information.

Configuring Automatic Prioritization for IP Phone Traffic

The **qos phones** command is used to enable or disable automatic prioritization of IP phone traffic. In addition, this command also specifies whether to trust the IP phone traffic (the default) or apply a specified priority value to the traffic. For example, the following command specifies a priority value to apply for ingress IP phone traffic:

```
-> qos phones priority 1
```

To trust IP phone traffic, enter the following command:

```
-> qos phones trusted
```

To disable automatic IP phone traffic prioritization for the switch, enter the following command:

```
-> qos no phones
```

Note that When automatic prioritization of IP phone traffic is enabled, QoS policies that specify priority are not applied to the IP phone traffic. Other QoS policies, however, are applied to this type of traffic as usual. If a policy specifies rate limiting, then the policy with the lowest rate limiting value is applied.

Prioritizing CPU Packets

In addition to physical switch ports, each NI has an internal CPU interface that handles traffic sent to or from the CPU (for example, BPDU and LAG PDUs). Such packets go directly to the CPU via a set of queues without traversing the switch fabric. In addition, packets from the CPU go directly to local ports without going through the fabric.

The QoS CPU priority policy action is used in a policy to assign a priority value to traffic destined for the CPU. See the **policy action cpu priority** command page in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information.

Prioritizing IEC 61850 Messages

AOS provides priority for different types of messages as per the IEC 61850. The Generic Object-Oriented Substation Event (GOOSE) messages and other associated message packets part of IEC 61850 getting switched through AOS switches can be applied with specific QoS priority.

The priority can be configured only for the supported IEC 61850 message types: goose, gse, sv, ptp, sntp, mms.

The priority can be set to "high", "medium", "low" or "default". High is mapped to QoS priority queue 7, medium is mapped to QoS priority queue 4, low is mapped to QoS priority queue 1. The "default" option can be used to restore to the default priority level.

The default priority for the IEC 61850 message types is:

Message Type	Default Priority
GOOSE	High
GSE	Medium
SV	High
MMS	Low
SNTP	Medium
PTP	Medium

The priority is applied by running a python script that programs the QoS rules for traffic prioritization.

The IEC 61850 message type rules condition, action, and rule names are predefined. The predefined names are reserved and cannot be used for user policy configuration.

IEC 61850 reserved rule names:

Function	Message Type	Condition	Action	Rule
1A. Trip	GOOSE	iec_goose_c	iec_goose_a	iec_goose_r
1B. Other	GSE	iec_gse_c	iec_gse_a	iec_gse_r
Raw Data	SV	iec_sv_c	iec_sv_a	iec_sv_r
Time Sync	PTP	iec_ptp_c	iec_ptp_a	iec_ptp_r
Time Sync	SNTP	iec_sntp_c	iec_sntp_a	iec_sntp_r
File Transfer	MMS	iec_mms_c	iec_mms_a	iec_mms_r

To configure the traffic prioritization for the IEC 61850 message type, use the **iec message-type priority** CLI command. For example:

```
-> iec message-type goose priority high
```

The message type can be one of the supported types: goose, gse, sv, ptp, sntp, mms, all.

A particular priority can be applied to all the message types at once by using the “all” option. For example:

```
-> iec message-type all priority high
```

The above example applies high priority to all the supported IEC 61850 message types.

The priority rule can be removed or flushed using the **iec message-type flush** CLI command. For example:

```
-> iec message-type goose flush
```

The configured priority rule can be removed or flushed from all the message types at once by using the “all” option. For example:

```
-> iec message-type all flush
```

The above example flushes all the rules applied for all the IEC 61850 message types.

The priority configured for the IEC 61850 message types can be viewed using the **iec show** CLI command. For example:

```
-> iec show
Message Type      Priority
---------------+------------
goose             high
sntp               low
```

Only the message types for which the priority is configured are displayed in the output.

Configuring Trusted Ports

By default switch ports are *untrusted*; that is, they do not recognize 802.1p or ToS/DSCP settings in packets of incoming traffic. When a port is untrusted, the switch sets the 802.1p or ToS/DSCP bits in incoming packets to the default 802.1p or DSCP values configured for that port.

The **qos port default 802.1p** and **qos port default dscp** commands are used to specify the default 802.1p and ToS/DSCP values. If no default is specified, then these values are set to zero.

Ports must be *both trusted and configured for 802.1Q* traffic in order to accept 802.1p traffic.

The following applies to ports that are trusted:

- The 802.1p or ToS/DSCP value is preserved.
- If the incoming 802.1p or ToS/DSCP flow does not match a policy, the switch places the flow into a default queue and prioritizes the flow based on the 802.1p or ToS/DSCP value in the flow.
- If the incoming 802.1p or ToS/DSCP flow matches a policy, the switch queues the flow based on the policy action.

The port trust setting can be configured globally or on a per-port basis to override the global setting.

To configure the global setting on the switch, use the **qos trust-ports** command. For example:

```
-> qos trust ports
```

To configure individual ports as trusted, use the **qos port trusted** command with the desired slot/port number. For example:

```
-> qos port 3/2/1 trusted
```

The global and port-level setting is active immediately; a **qos apply** is not required to activate the change. See "Applying the Configuration" on page 28-72 for more information.

To display information about QoS ports, such as whether or not the port is trusted, use the **show qos port** command. For example:

```
-> show qos port
Slot/                  Default    Default                Bandwidth           DEI
Port    Active  Trust P/DSCP Classification  Physical  Ingress Egress    Map Mark   Type
-------+-------+-----+------+--------------+----------+-------+------+------+------+---------
1/1/1      No      No  0/ 0           DSCP         0K       -      -     No    No   ethernet
1/1/2      No    *Yes  0/ 0        *802.1P         0K       -      -     No    No   ethernet
1/1/3      No      No  0/ 0           DSCP         0K       -      -     No    No   ethernet
1/1/4      No      No  0/ 0           DSCP         0K       -      -     No    No   ethernet
1/1/5      No      No  0/ 0           DSCP         0K       -      -     No    No   ethernet
1/1/6      No      No  0/ 0           DSCP         0K       -      -     No    No   ethernet
1/1/7      No      No  0/ 0           DSCP         0K       -      -     No    No   ethernet
1/1/8      No      No  0/ 0           DSCP         0K     50K      -     No    No   ethernet
```

Using Trusted Ports With Policies

Whether or not the port is trusted is important if you want to classify traffic with 802.1p bits. If the policy condition specifies 802.1p, the switch must be able to recognize 802.1p bits. (Note that the trusted port must also be 802.1Q-tagged). The 802.1p bits can be set or mapped to a single value using the **policy action 802.1p** command.

In the following example, the **qos port** command specifies that port 2 on slot 3 are able to recognize 802.1p bits. A policy condition (**Traffic**) is then created to classify traffic containing 802.1p bits set to 4 and destined for port 2 on slot 3. The policy action (**SetBits**) specifies that the bits are reset to 7 when the traffic egresses the switch. A policy rule called **Rule2** puts the condition and the action together.

```
-> qos port 3/2/1 trusted
-> policy condition Traffic destination port 3/2 802.1p 4
-> policy action SetBits 802.1p 7
-> policy rule Rule2 condition Traffic action SetBits
```

To activate the configuration, enter the **qos apply** command. See "Applying the Configuration" on page 28-72 for more information.

For actions that set 802.1p bits, note that a limited set of policy conditions are supported. See "Condition and Action Combinations" on page 28-33 for more information.

Note. 802.1p mapping can also be set for Layer 3 traffic, which typically has the 802.1p bits set to zero.

Congestion Management

Queuing mechanisms are used to manage congestion on egress ports. When congestion occurs, packets are prioritized and placed into queues based on the CoS markings assigned to the packets during classification. If there is no congestion on the egress port, packets are sent out as soon as they are received.

There are eight egress queues allocated for each port on an OmniSwitch. Queue Set Profiles (QSPs) or Data Center Bridging (DCB) profiles are used to provide traffic management and QoS scheduling for the egress queues. For more information, see "Queue Sets" on page 28-11 and "QSet Profiles" on page 28-13.

Queue Sets

The queue management and related QoS functions are implemented using a framework based on Queue Sets (QSets). A QSet is a set of eight egress port queues that are associated with each switch port.

The QSET framework involves the following elements:

- **QSet instance (QSI)**—A QSI is a logical entity that refers to a set of eight queues. Each port in the switch is automatically associated with a QSI.
- **QSet profile (QSP)**—a profile associated with each QSI that defines the output scheduling behavior for the queues associated with the QSet instance. See "QSet Profiles" on page 28-13.
- **Data Center Bridging profile (DCP)**—a profile associated with each QSI that defines the output scheduling behavior for the queues associated with the QSet instance. *Note that Data Center Bridging is supported only on the OmniSwitch 6900.*

  There are eleven predefined DCPs, with DCP 8 serving as the default profile that is automatically assigned to each QSI. DCB profiles are only applied when the switch is using the OmniSwitch Data Center license. See the "Configuring Data Center Bridging" chapter in the *OmniSwitch AOS Release 8 Data Center Switching Guide.*

Note. QSet profiles and DCB profiles are mutually exclusive. If the OmniSwitch Data Center software license is installed, then DCB profiles are used. If this license is not installed, then QSet profiles are used.

How it Works

When a physical switch port comes up, a QSet instance (a set of eight queues) is automatically associated with the port for unicast traffic. In addition, the default QSet profile (QSP 1) or the default DCB profile (DCP 8) is automatically assigned to the QSI.

If a port attaches to a link aggregate (LAG), a QSI and default QSP 1 or default DCP 8 are automatically associated with the LAG ID. Each time a port joins the LAG, the QSI for the port is imported into the LAG. When this occurs, the LAG QSI becomes the parent and the member port QSI is the child. Note that when a member port leaves a LAG, the QSI and profile for the port reverts back to the default values.

The following example diagram shows the relationship between switch ports, QSet instances, and QSet profiles as they apply to unicast traffic. See "QSet Profiles" on page 28-13 for more information.

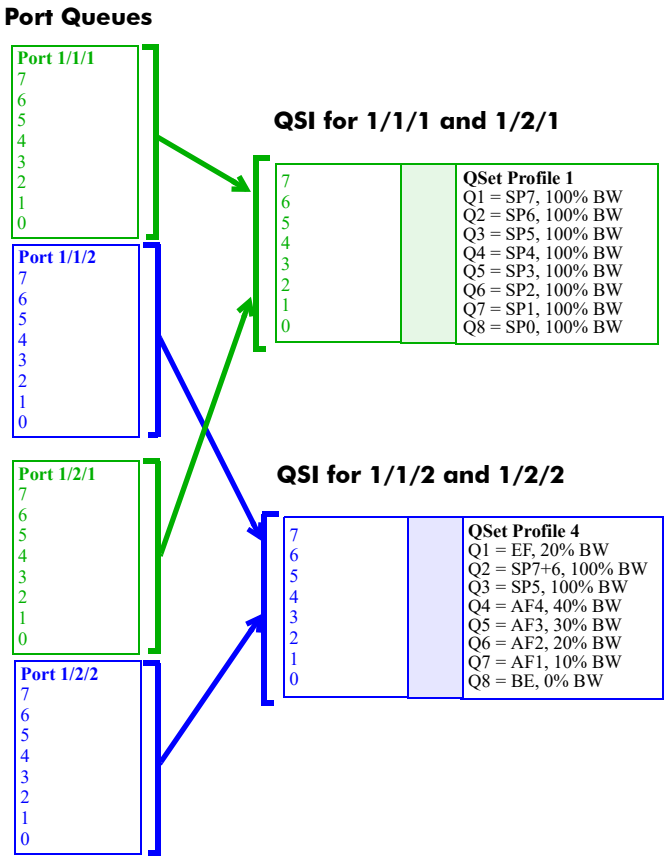


Figure 28-2 : Queue Set (QSet) Framework (Unicast Traffic)

In this example illustration:

- The switch maintains a QSI for each port (1/1/1, 1/1/2, 1/2/1, and 1/2/2).
- A QSP is applied to the QSI for a port when the QSP is assigned to the port.
- There are predefined QSPs available on each switch. In this example, the default QSP 1 is associated with the QSI for port 1/1/1 and port 1/2/1. However, QSP 4 was assigned to the QSI for port 1/2/1 and port 1/2/2.
- The QSet framework shown in this example applies to unicast traffic. Selecting QSPs only applies to unicast queue management.

Multicast Queues

Unicast and multicast traffic are both queued and funneled separately through the switch. The QSet framework described in previous sections applies only to unicast traffic. Multicast traffic is queued based on the destination multicast group ID (MGID) for the packets. Pre-set queues and profiles associated with the MGID handle the multicast traffic.

The multicast queue framework is not user-configurable in that there are no user-configurable profiles. However, the type of profile assigned to a port can determine the class of service for multicast traffic. For more information, see “Multicast and Unicast Traffic Distribution” on page 28-17.

QSet Profiles

The following QSet profiles (QSPs) are supported on the specified switch platforms:

OmniSwitch	QSP1 (All-SP)	QSP2	QSP3	QSP4	QSP5 (All-WRR)	Custom QSP	Base QSP for Custom QSP
OS6360	Yes	Yes	No	No	No	Yes	QSP1, QSP2
OS6465	Yes	No	No	No	Yes	Yes	QSP1, QSP5
OS6560	Yes	No	No	No	Yes	Yes	QSP1, QSP5
OS6860/6865	Yes	Yes	Yes	Yes	Yes	Yes	QSP1, QSP5
OS6860N*	Yes	Yes	No	No	No	Yes	QSP1, QSP2
OS6900	Yes	Yes	Yes	Yes	No	No	N/A
OS6900-X72	Yes	Yes	Yes	Yes	Yes	Yes	QSP1, QSP5
OS6900-V72/C32*	Yes	Yes	No	No	No	Yes	QSP1, QSP2
OS6900-X48C6/T48C6/ X48C4E/V48C8*	Yes	Yes	No	No	No	Yes	QSP1, QSP2
OS9900	Yes	No	No	No	Yes	Yes	QSP1, QSP5

* QSP2 has all WRR schedulers.

Each profile defines the following bandwidth management attributes that are applied to traffic destined for the port or LAG QSet instance associated with the profile:

- The percentage of bandwidth allocated for and shared by all of the QSet queues. This value is taken from the port to which the QSet profile is applied (either port speed or the user-defined bandwidth for the port is used).
- The administrative status of statistics collection for the QSet queues.
- The queue specific (QSpec) priority used for output scheduling on each of the eight QSet queues.

To determine how flows are mapped to the egress queues based on ingress priority markings, see the "QSet Profile Mapping (Unicast)" on page 28-15. This section contains CoS priority mapping tables for each QSet profile.

Configuring QSet Profiles

The default QSet profile (QSP 1) is automatically assigned to each QSet instance when a port goes active or a port joins a LAG. It is only necessary to assign a different profile if QSP 1 attributes are not sufficient.

Consider the following when configuring a QSet profile:

- QSP 1, 2, 3, 4, and 5 are predefined profiles that are not modifiable and cannot be deleted from the switch configuration.
- Creating a new custom profile is allowed by importing one of the predefined profiles into a new profile ID between 6 and 16, then modifying the new profile attributes as necessary. See "Creating a Custom Profile" on page 28-14 for more information.

- There is only one QSP assigned to each QSet instance and only one QSet instance per port or link aggregate (LAG). However, a LAG may show multiple QSet instances, one for each port that is a member of the LAG.
- When a port leaves a LAG, the default QSP 1 profile is associated with the QSet instance for that port. In other words, if the QSet instance for a port was associated with QSP 4 when the port joined the LAG, the port is associated with QSP 1 when it leaves the LAG.

The **qos qsi qsp** command is used to change the QSP for a specific QSet instance (QSI). For example:

```
-> qos qsi port 1/2/1 qsp 2
-> qos qsi port 1/2/2-10 qsp 3
-> qos qsi linkagg 5 qsp 3
```

To view the QSet profile configuration for the switch, use the **show qos qsp** command.

See the "QoS Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the **qos qsi qsp** and related **show** commands.

Changing the Default QSet Profile

Consider the following when changing the default QSet profile for the switch:

- Only the OmniSwitch 6860 and OmniSwitch 6865 support changing the default system profile.
- The new default profile is applied to all switch ports and link aggregates.
- A QSP assigned through the **qos qsi qsp** command overrides the system default QSP assignment. For example, if the system default QSP is set to 1 and the **qos qsi qsp** command is used to change the QSP to 2 on port 1/1/20, the QSP 2 settings are applied to port 1/1/20.

To change the default QSet profile (QSP 1) to one of the other supported profiles (QSP 2, 3, or 4), use the **qos qsp system-default** command. For example, the following command changes the default profile to QSP 2:

```
-> qos qsp system-default 2
```

To view the current default QSet profile setting for the switch, use the **show qos qsp system-default** command.

Creating a Custom Profile

A custom profile is created by importing a predefined QSet profile to use as a template for the custom profile. As the table shows in "QSet Profiles" on page 28-13, predefined QSP 1, 2, or 5 is used as a custom profile template based on the supported switch platform.

The **qos qsp import** command is used to create a custom profile. For example, the following command creates profile 6 (QSP 6) using predefined QSP 5 as the template for the new profile:

```
-> qos qsp 6 import qsp 5
```

Once the custom QSet profile is created, the following attributes for each queue profile (QP) are configurable using the **qos qsp qp** command:

QP Attribute	Command Parameters	Description
Committed Information Rate	**cir** %	Configures the Committed Information Rate that is applied to queue traffic.

QP Attribute	Command Parameters	Description
Peak Information Rate	**pir** *%*	Configures the Peak Information Rate limit that is applied to queue traffic.
Queue weight	**weight** *1-100*	Configures the weight assigned to the individual queue. The higher the weight value, the more shared bandwidth that is allocated for that queue.

Consider the following guidelines regarding queue profile attributes for a custom QSet profile:

- Configuring the Committed Information Rate is not supported on the OmniSwitch 6465, OmniSwitch 6560, and OmniSwitch 9900.
- The granularity of queue bandwidth shaping should not be less than 10% of the port bandwidth.
- When a QSet profile specifies a Weighted Round Robin (WRR) scheduler, bandwidth is shared among all of the queues that are receiving traffic. The weight value assigned to each queue determines the percentage of bandwidth allocated for that queue. For example, if three queues are receiving traffic and all have the same weight value, the bandwidth is divided up evenly between the queues. If the weight value is increased for one of these queues, that queue will receive a higher percentage of bandwidth.
- Configuring a Committed Information Rate for a WRR QSet profile is redundant, as the WRR scheduler guarantees a minimum bandwidth for each queue.
- Changing the queue scheduler for a custom QSet profile is not supported.
- When a custom profile is modified, the changes are applied to all ports that are associated with that custom profile. To apply specific changes to a single port (QSet instance), import a custom or predefined profile into a new custom profile, make the necessary changes, then apply the new custom profile to the port.

QSet Profile Mapping (Unicast)

This sections contains a unicast queue mapping table for each of the predefined QSet profiles (QSPs). By default, each QSet port instance is associated with QSP 1. To determine which QSPs are supported on specific switch platforms, see "QSet Profiles" on page 28-13.

Default QSet Profile 1 (8 SP)

Queue ID	Queue Type	Scheduling	Weight	802.1p	ToS	DSCP	Notes
1	SP7	SP	100%	7	7	7.x	Straight SP7
2	SP6	SP	100%	6	6	6.x	Straight SP6 with starvation
3	SP5	SP	100%	5	5	5.x, 5.6	Straight SP5 with starvation ("unprotected" EF)
4	SP4	SP	100%	4	4	4.x	Straight SP4 with starvation
5	SP3	SP	100%	3	3	3.x	Straight SP3 with starvation
6	SP2	SP	100%	2	2	2.x	Straight SP2 with starvation
7	SP1	SP	100%	1	1	1.x	Straight SP1 with starvation
8	SP0	SP	100%	0	0	0	Straight SP0 with starvation

QSet Profile 2 (1 EF + 7 SP)

Note. On the OmniSwitch 6860N, OmniSwitch 6900-V72/C32 and OmniSwitch 6900-X48C6/T48C6, QSet Profile 2 maps WRR scheduling to all 8 queues.

Queue ID	Queue Type	Scheduling	Weight	802.1p	ToS	DSCP	Notes
1	EF	SP	20%	X(5)	X(5)	5.6	Protected EF
2	SP7+SP6	SP	100%	7, 6	7, 6	7.x, 6.x	Straight SP 7 and 6 max (effective CIR = PR minus EF PIR)
3	SP5	SP	100%	5	5	5.x	Straight SP5 with starvation
4	SP4	SP	100%	4	4	4.x	Straight SP4 with starvation
5	SP3	SP	100%	3	3	3.x	Straight SP3 with starvation
6	SP2	SP	100%	2	2	2.x	Straight SP2 with starvation
7	SP1	SP	100%	1	1	1.x	Straight SP1 with starvation
8	SP0	SP	100%	0	0	0	Straight SP0 with starvation

QSet Profile 3 (1 EF + 7 WFQ)

Queue ID	Queue Type	Scheduling	Weight	802.1p	ToS	DSCP	Notes
1	EF	SP	20%	X(5)	X(5)	5.6	Protected EF
2	WFQ7+6	WFQ	20%	7, 6	7, 6	7.x, 6.x	WFQ
3	WFQ5	WFQ	12%	5	5	5.x	WFQ
4	WFQ4	WFQ	12%	4	4	4.x	WFQ
5	WFQ3	WFQ	12%	3	3	3.x	WFQ
6	WFQ2	WFQ	38%	2	2	2.x	WFQ
7	WFQ1	WFQ	4%	1	1	1.x	WFQ
8	WFQ0	WFQ	2%	0	0	0	WFQ

QSet Profile 4 (1 EF + 2 SP + 4 AF + 1 BE)

Queue ID	Queue Type	Scheduling	Weight	802.1p	ToS	DSCP	Notes
1	EF	SP	20%	X(5)	X(5)	5.6	Protected EF
2	SP7+6	SP	100%	7, 6	7, 6	7.x, 6.x	SP 7 with effective CIR = PR minus EF PIR
3	SP5	SP	100%	5	5	5.x	SP 6 with effective CIR = PR minus EF PIR (starvable) “Mission Critical” data/video
4	AF4	WFQ	40%	x	x	4.1, 4.2, 4.3	AF4 WFQ (starvable)
5	AF3	WFQ	30%	x	x	3.1, 3.2, 3.3	AF3 WFQ (starvable)
6	AF2	WFQ	20%	x	x	2.1, 2.2, 2.3	AF2 WFQ (starvable)
7	AF1	WFQ	10%	x	x	1.1, 1.2, 1.3	AF1 WFQ (starvable)

Queue ID	Queue Type	Scheduling	Weight	802.1p	ToS	DSCP	Notes
8	BE	WFQ	0%	4, 3, 2, 1, 0	4, 3, 2, 1, 0	4.0, 3.0, 2.0, 1.0, 0.0	BE not guaranteed

QSet Profile 5 (8 WRR)

Queue ID	Queue Type	Scheduling	Weight	802.1p	ToS	DSCP	Notes
1	WRR7	WRR	1	7	7	7.x	WRR7 shares bandwidth with other queues receiving traffic.
2	WRR6	WRR	1	6	6	6.x	WRR6 shares bandwidth with other queues receiving traffic.
3	WRR5	WRR	1	5	5	5.x,	WRR5 shares bandwidth with other queues receiving traffic.
4	WRR4	WRR	1	4	4	4.x	WRR4 shares bandwidth with other queues receiving traffic.
5	WRR3	WRR	1	3	3	3.x	WRR3 shares bandwidth with other queues receiving traffic.
6	WRR2	WRR	1	2	2	2.x	WRR2 shares bandwidth with other queues receiving traffic.
7	WRR1	WRR	1	1	1	1.x	WRR1 shares bandwidth with other queues receiving traffic.
8	WRR0	WRR	1	0	0	0	WRR0 shares bandwidth with other queues receiving traffic.

Multicast and Unicast Traffic Distribution

The following Class of Service (CoS) model for unicast and multicast traffic is applied when either the default QSet profile (QSP 1) or the default Data Center Bridging (DCB) profile 8 is the active profile for the port.

Cos 0 - Lower Priority MC (0-3) = 10

Cos 1 - Higher Priority MC (4-7) = 52

Cos 3 - All Other Unicast UC(0-7) = 108

Cos 7 - CPU Generated Packets = 127 (maximum weight)

For example:

- When sending two streams of 100% MC Lower Priority and 100% MC Higher Priority, the distribution should be 10 and 50 packets, which is approximately 17% of Lower Priority MC and 83% of Higher Priority.
- When sending Lower Priority MC 100% and UC 100%, the distribution is 9% of MC and 91% of UC.
- When sending Higher Priority MC 100% and UC 100%, the distribution is 32% of MC and 68% of UC.

For information about multicast and unicast traffic distribution on specific OmniSwitch 6900 models, see “Multicast/Unicast Traffic Distribution for the OmniSwitch 6900-X72/Q32” on page 28-19.

Non-Default Profile

The CoS model implemented also applies for non-default QSet profiles (QSP 2–4), except on the OmniSwitch 6900. The multicast and unicast queue mapping for non-default QSet profiles (QSP 2–4) and non-default DCB profiles (DCP 1–7, 9–128) on the OmniSwitch 6900, is as follows:

Strict Priority Profiles (for example, DCP 7)

Queues	Priority	Precedence
UC7	7	Highest
MC3	7, 6	
UC6	6	
UC5	5	
MC2	5, 4	
UC4	4	
UC3	3	
MC1	3, 2	
UC2	2	
UC1	1	
MC0	1, 0	
UC0	0	Lowest

Weighted Round Robin (WRR) Profiles

Queues	Priority	Weight
UC7	7	W7
MC3	7, 6	Avg(W7,W6)
UC6	6	W6
UC5	5	W5
MC2	5, 4	Avg(W5,W4)
UC4	4	W4
UC3	3	W3
MC1	3, 2	Avg(W3,W2)
UC2	2	W2
UC1	1	W1
MC0	1, 0	Avg(W1,W0)
UC0	0	W0

Note: Wn = Weight of UCn
Avg(Wn,Wm) = Average of Weights of UCn & UCm

Profile with a Mix of Strict Priority and WRR

Unicast queues configured as Strict Priority will inherit behavior from the Strict Priority model, and unicast queues configured as WRR will inherit behavior from the WRR model. Multicast queues will always follow the behavior that the corresponding unicast queues are following. For example:

- If UC7 and UC6 are Strict Priority, then the MC3 (priority 6 and 7) will also use Strict Priority.
- If UC7 and UC6 are Weighted Round Robin, then MC3 (priority 6 and 7) will also use Weighted Round Robin. The weight of MC3 will be the average of the weights for UC6 and UC7.

For DCB profile ETS behavior, where a Traffic Class (TC) can have more than one priority, multicast queues will follow the corresponding unicast queue behavior. For example:

DCB Profile 1:

TC 0 Priority 0 -3)
TC 1 Priority 4 -5)
TC 2 Priority 6 -7)

TC 0 has UC0 through UC3 in Round Robin, so MC0 (priority 0 and 1) and MC1 (priority 2 and 3) will also participate in the Round Robin behavior of TC 0.

Multicast/Unicast Traffic Distribution for the OmniSwitch 6900-X72/Q32

The following table shows the multicast and unicast queue mapping for the OmniSwitch 6900-X72/Q32:

Queues	Priority	Precedence
UC7, MC7	7	Highest
UC6, MC6	6	
UC5, MC5	5	
UC4, MC4	4	
UC3, MC3	3	
UC2, MC2	2	
UC1, MC1	1	
UC0, MC0	0	Lowest

There are additional multicast queues in both switches. As a result, traffic distribution is based on the priority value regardless of whether the traffic is multicast or unicast. In other words, multicast traffic with priority 1 and unicast with priority 1 are given equal distribution.

Multicast Source PFC on the OmniSwitch 6900

Ingress admission control on the OmniSwitch 6900 does not distinguish between unicast and multicast traffic. Therefore, a multicast source connected to a port which is PFC aware will react to congestion thereby pausing transmission. This will affect multicast hosts not in the congestion path.

When a multicast source is attached to a port on a OmniSwitch 6900, make sure that PFC is not enabled for that particular priority on the ingress. This can be done by configuring the port to use DCP 8 (all priorities are lossless) or for instance, DCP 1 (priority 4 and 5 are lossless, so multicast may be sent at any other priority other than priority 4 or 5).

If multicast sources are configured to react to PFC, it will affect subscribers not in the congestion path.

OmniSwitch Congestion Avoidance

Congestion avoidance mechanisms monitor queues to provide early detection and notification of potential queue congestion. If necessary, such mechanisms may even strategically drop low priority (non-conforming) packets to prevent congestion. Dropping packets signals the packet source to decrease the transmission rate, thus preventing the queue from overflowing.

A packet is color marked during the QoS classification process to indicate a drop precedence for the packet.

- Green = Committed
- Yellow = Conformed
- Red = Exceeded

Color marking techniques supported include Single-Rate Tri-Color Marking (srTCM) and Two-Rate Tri-Color Marking (trTCM). TCM is applied to ingress traffic using a QoS policy rule (see "Tri-Color Marking" on page 28-22 for more information). Note that all packets that are not marked with a specific color are treated as green (committed) packets.

When congestion of green, yellow, and red traffic occurs, traffic within the same priority is processed as follows:

- On the OmniSwitch 6900, green has the highest precedence and red and yellow are dropped. When congestion of yellow and red traffic occurs, yellow and red have the same precedence.
- On the OmniSwitch 6465, 6860, 6860N, 6865, OmniSwitch 6900-V72/C32, OmniSwitch 6900-X48C6/T48C6, and OmniSwitch 9900, the green, yellow, and red traffic is given equal precedence.

Traffic Policing and Shaping

Traffic policing and shaping mechanisms are used to limit the rate of traffic. The main difference between the two is how they handle traffic that violates the specified rate. Policing either drops or remarks traffic that exceeds a configured maximum rate. Shaping delays the transmission of packets that exceed configured rates by placing the packets in a queue and scheduling them to be sent at a later time.

The OmniSwitch provides the following techniques for policing and shaping traffic flows.

Policing

- **QoS Tri-Color Marking (TCM) policy**. A TCM policy consists of a policy action that specifies packet rates and burst sizes. The policy condition defines the type of traffic for TCM to meter and then color mark (green, yellow, or red) based on conformance with the rate limits defined in the policy action. See "Tri-Color Marking" on page 28-22.
- **QoS bandwidth policy actions**. Maximum bandwidth and depth policy actions are used in QoS policy rules to specify a maximum ingress bandwidth rate and bucket size. See "Configuring Policy Bandwidth Policing" on page 28-25 for more information.
- **Port-based QoS bandwidth shaping**. The QoS CLI provides three commands for setting the maximum ingress and egress bandwidth rate and bucket size for a specific port. These QoS port parameters define the rate at which traffic is received and sent on the specified port. See "Configuring Port Bandwidth Policing" on page 28-27.
- **E-Services bandwidth parameters**. The VLAN Stacking Service Access Point (SAP) profile defines an ingress and egress bandwidth rate limiting configuration for an Ethernet Service. See Chapter 37, "Configuring VLAN Stacking," for more information.
- **Universal Profile (UNP) bandwidth parameters.** A UNP profile defines an ingress and egress bandwidth rate limiting configuration for device ports that are assigned to the profile. See Chapter 30, "Configuring Access Guardian," for more information.

Shaping

Queue bandwidth shaping. This type of shaping is implemented through the queue management architecture of the switch. A set of eight egress queues for each port (QSet) is associated with a profile that defines and applies the shaping and scheduling configuration for each queue in the QSet. See "Congestion Management" on page 28-11 for more information.

Tri-Color Marking

This implementation of a Tri-Color Marking (TCM) provides a mechanism for policing network traffic by limiting the rate at which traffic is sent or received on a switch interface. The TCM policier meters traffic based on user-configured packet rates and burst sizes and then marks the metered packets as green, yellow, or red based on the metering results.

The following diagram illustrates the basic operation of TCM:

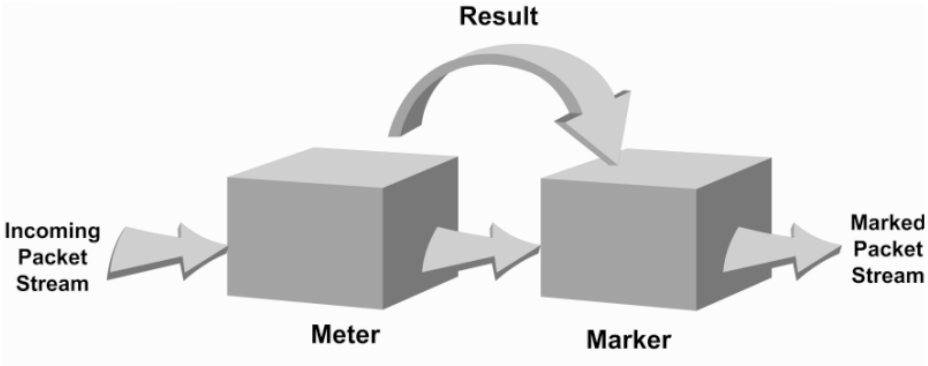


Figure 28-3 : Basic Operation of Tri-Color Marking

The TCM policier meters each packet and passes the metering result along with the packet to the Marker. Depending upon the result sent by the Meter, the packet is then marked with either the green, yellow, or red color. The marked packet stream is then transmitted on the egress based on the color-coded priority assigned.

The TCM Meter operates in Color-Blind mode (the Color-Aware mode is not supported). In the Color-Blind mode, the Meter assumes that the incoming packet stream is uncolored.

There are two types of TCM marking supported:

- Single-Rate TCM (srTCM)—Packets are marked based on a Committed Information Rate (CIR) value and two associated burst size values: Committed Burst Size (CBS) and Peak Burst Size (PBS).
- Two-Rate TCM (trTCM)—Packets are marked based on a CIR value *and* a Peak Information Rate (PIR) value and two associated burst size values: CBS and PBS.

Both srTCM and trTCM operate in the same basic manner, as shown in the above diagram. The main difference between the two types is that srTCM uses one rate limiting value (CIR) and trTCM uses two rate limiting values (CIR and PIR) to determine packet marking.

The type of TCM used is determined when the policier is configured; depending on which rates and burst size values are configured, TCM functions in ether single-rate or two-rate mode. There is no explicit command to select the type of TCM. See "Configuring Tri-Color Marking" on page 28-23 for more information.

Based on the TCM type used, packets are marked as follows:

TCM Type	Meter Compliance	Marker Color	Result
Single-Rate (srTCM)	Packet is CIR/CBS compliant.	GREEN	Packet is transmitted with the Drop Precedence set to LOW.

TCM Type	Meter Compliance	Marker Color	Result
	Packet is not CIR/CBS compliant but is CIR/PBS compliant.	YELLOW	Packet is transmitted with the Drop Precedence set to HIGH (packet is dropped first when congestion occurs on the egress queue.
	Packet is neither CIR/CBS nor CIR/PBS compliant.	RED	Packet is dropped at the ingress.
Two-Rate (trTCM)	Packet is CIR/CBS compliant.	GREEN	Packet is transmitted with the Drop Precedence set to LOW.
	Packet is not CIR/CBS compliant but is PIR/PBS compliant.	YELLOW	Packet is transmitted with the Drop Precedence set to HIGH (packet is dropped first when congestion occurs on the egress queue.
	Packet is neither CIR/CBS nor PIR/PBS compliant.	RED	Packet is dropped at the ingress.

Configuring Tri-Color Marking

Tri-Color Marking (TCM) is a supported technique for policing traffic. See "Tri-Color Marking" on page 28-22 for an overview of how this implementation of TCM works.

Configuring TCM is done by creating a TCM policy action using the following QoS **policy action** command parameters:

- **cir** (Committed Information Rate, in bits per second)
- **cbs** (Committed Burst Size, in bytes)
- **pir** (Peak Information Rate, in bits per second)
- **pbs** (Peak Burst Size, in bytes)
- **color-only** (mark packet color only)

Consider the following when configuring TCM policy actions:

- There is no explicit CLI command to specify the mode in which the TCM meter operates. This mode is determined by whether or not the PIR is configured for the policy action and if the value of the PIR is greater than the value of the specified CIR. In this case, the trTCM mode is triggered; otherwise, the srTCM mode is used.
- This implementation of TCM is in addition to the basic rate limiting capabilities provided through the maximum bandwidth and maximum depth parameters used in QoS policy actions and the ingress and egress bandwidth parameters used in VLAN Stacking Service Access Point (SAP) profiles. When these parameters are used, the TCM meter operates in the Single-Rate TCM mode by default.
- A srTCM policy action specifies both a CBS and PBS value. Default values for these burst sizes are used if one is not specified using the optional **cbs** and **pbs** parameters.
- Configure the PBS and CBS with a value that is greater than or equal to the size of the largest IP packet in the metered stream.

To configure a TCM QoS policy action, use the **policy action cir** command with one or more of the above parameters. Configuring the **cbs** and **pbs** parameters is optional. If a value is not specified for either one, the default value is used for both parameters. For example:

```
-> policy action A1 cir 10M
```

To specify one or both of the burst size values, use the **cbs** and **pbs** parameters. For example:

```
-> policy action A2 cir 10m cbs 4k
-> policy action A3 cir 10m cbs 4k pbs 10m
```

All of these command examples configure the TCM meter to operate in the Single-Rate TCM (srTCM) mode. To configure the meter to operate in the Two-Rate TCM (trTCM) mode, use the **pir** parameter and specify a peak information rate value that is greater than the committed information rate value. For example, the following commands configure the meter to use the trTCM mode:

```
-> policy action A4 cir 10m cbs 4k pir 20m
-> policy action A5 cir 10m cbs 4k pir 20m pbs 40m
```

Once a TCM policy action is configured, the action can be used in a policy rule to rate limit traffic according to the specified rates and burst sizes. Traffic that matches a TCM policy is marked green, red, or yellow based on the rate limiting results.

To remove the TCM configuration from a QoS policy action, use the **no** form of the **policy action cir** command. For example:

```
-> policy action A6 no cir
```

TCM Policy Example

Once configured, a TCM policy action is then available to use in a QoS policy rule to apply color marking to a specified traffic stream.

First, create a condition for the traffic. In this example, the condition is called **ip_traffic**. A policy action (**tcm1**) is then created to enforce ingress rate limiting using TCM.

```
-> policy condition ip_traffic source ip 10.10.5.3
-> policy action tcm1 cir 5m cbs 4k pir 10m pbs 20m counter-color green-nongreen
-> policy rule rule1 condition ip_traffic action tcm1
```

Note that the rates and burst sizes can be specified in abbreviated units, in this case, **10m**.

The rule is not active on the switch until the **qos apply** command is entered. When the rule is activated, any flows coming into the switch from source IP address 10.10.5.3 is metered and marked according to the TCM policier parameters specified in the **tcm1** policy action.

Setting the DEI Bit

The Drop Eligible Indicator (DEI) bit setting is applied to packets marked yellow (non-conforming) as the result of Tri-Color Marking (TCM) or other rate limiting mechanisms. The TCM policier meters traffic based on user-configured packet rates and burst sizes and then marks the metered packets as green, yellow, or red based on the metering results. See "Configuring Tri-Color Marking" on page 28-23 for more information.

Yellow packets are assigned a high drop precedence, which means they are dropped first when the egress port queues become congested. If there is no congestion on the queues, however, yellow packets are retained and forwarded along to the next switch. When this occurs, the receiving switch does not know that the packet was marked yellow by the transmitting switch.

Setting the DEI bit for yellow egress packets ensures that the upstream switch is made aware that the packet was marked yellow. The upstream switch can then decide to drop the DEI marked packets first when the network is congested. When a switch receives a yellow packet with the DEI bit set and DEI mapping is enabled, the packet is mapped to an internal drop precedence or yellow color marking for the switch.

The switch can be set globally so that DEI bit marking and mapping is enabled for all ports. Individual ports can be configured to override the global setting

Configuring the DEI Bit Setting

By default, DEI bit marking (egress) and mapping (ingress) is disabled on all switch ports. The DEI bit setting operation can be configured globally on the switch or on a per-port basis.

To configure the global DEI bit setting operation to mark traffic egressing on QoS destination ports, use the **qos dei** command with the **egress** parameter option. For example:

```
-> qos dei egress
```

To configure the switch to map ingress traffic marked with the DEI bit, use the **qos dei** command with the **ingress** parameter option. For example:

```
-> qos dei ingress
```

To configure the DEI bit operation for an individual port, use the **qos port dei** with the **ingress** or **egress** parameter option. For example:

```
-> qos port 1/1/10 dei egress
-> qos port 1/1/11 dei ingress
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about these commands.

Configuring Policy Bandwidth Policing

The **policy action maximum bandwidth** and **policy action maximum depth** commands are used to configure QoS policy actions. Both actions are typically used in combination; the bucket size (depth) determines how much over the maximum bandwidth the traffic can burst.

The maximum bandwidth and maximum depth actions are configured as part of a QoS policy in which the condition specifies the type of traffic to rate limit. Maximum bandwidth policies are applied to source (ingress) ports and/or flows. See the "Bandwidth Policing Example" on page 28-77.

Port Groups and Maximum Bandwidth

If a port group condition (see "Creating Port Groups" on page 28-59) is used in a maximum bandwidth policy, the bandwidth value specified is shared across all ports in the group.This also applies to flows that involve more than one port. For example, if a policy specifies a maximum bandwidth value of 10M for a port group containing 4 ports, the total bandwidth limit enforced is 10M for all 4 ports.

Note the following when configuring ingress maximum bandwidth policies:

- If a policy condition applies to ports that are located on different slots, the maximum bandwidth limit specified is multiplied by the number of slots involved. For example, if a rule is configured to apply a maximum bandwidth limit of 10M to ports 1/1/1, 1/3/10, and 1/4/5, then the actual bandwidth limit enforced for all three ports is 30M.

- On an OmniSwitch 9900, metering is defined per switch ASIC. For example, if a rule is configured to apply a maximum bandwidth limit of 10M to ports 1/3/7 and 1/3/20 and each port is on a different ASIC, then the metering is performed for each port resulting in 20M rate-limiting instead of 10M. If both ports were on the same ASIC, then 10M rate-limiting would apply.
- The maximum traffic received by a destination port is also dependent on how many slots are sending traffic to the destination port. However, each slot is restricted to sending only 10k.
- If a policy condition applies to ports that are all on the same slot, then the maximum bandwidth value specified in the rule is not increased.
- Ingress bandwidth limiting is done using a granularity of 64K bps.
- The **show active policy rule** command displays the number of packets that were dropped because they exceeded the ingress bandwidth limit applied by a maximum bandwidth policy.
- Although bandwidth policies are applied to ingress ports, it is possible to specify a destination port or destination port group in a bandwidth policy as well. Doing so, effects egress rate limiting/egress policing on the ingress port itself.

The following subsections provide examples of ingress maximum bandwidth policies using both source and destination port groups.

Example 1: Source Port Group

In the following example, a port group (**pgroup**) is created with two ports and attached to a policy condition (**Ports**). A policy action with maximum bandwidth is created (**MaxBw**). The policy condition and policy action are combined in a policy rule called **PortRule**.

```
-> policy port group pgroup 1/1/1-2
-> policy condition Ports source port group pgroup
-> policy action MaxBw maximum bandwidth 10k
-> policy rule PortRule condition Ports action MaxBw
```

In this example, if both ports 1 and 2 are active ports, the 10000 bps maximum bandwidth is shared by both ports. In other words, maximum bandwidth policies for port groups define a maximum bandwidth value that is a total bandwidth amount for all ports, not an amount for each port.

Example 2: Destination Port Group

In the following example, a port group (**pgroup2**) is created with several ports and attached to a policy condition (**Ports2**). A policy action with maximum bandwidth is created (**MaxBw**). The policy condition and policy action are combined in a policy rule called **PortRule2**.

```
-> policy port group pgroup2 1/1/1 1/1/25 1/2/1
-> policy condition Ports2 destination port group pgroup2
-> policy action MaxBw maximum bandwidth 10k
-> policy rule PortRule2 condition Ports2 action MaxBw
```

In this example, the specified ports for **pgroup2** span across two slots. As a result, the maximum bandwidth limit specified by the policy action is increased to 20K for all of the ports. The bandwidth limit is increased by multiplying the number of slots by the specified bandwidth value.

Configuring Port Bandwidth Policing

QoS supports configuring maximum bandwidth on ingress and egress ports through the **qos port maximum egress-bandwidth**, **qos port maximum ingress-bandwidth**, and **qos port maximum depth** CLI commands. For more information about these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Note the following when configuring the ingress or egress bandwidth limit for a port:

- Maximum bandwidth limiting is done using a granularity of 64K bps. Any value specified that is not a multiple of 64K is rounded up to the next highest multiple of 64K.
- The maximum bandwidth value cannot exceed the maximum bandwidth of the interface type associated with the port.
- Modifying the maximum bandwidth is most useful for low-bandwidth links.
- The configured port-based egress bandwidth limit takes precedence over an egress queue limit configured on the same port.

Configuring Maximum Egress Bandwidth

Configure the maximum rate at which traffic is sent on the specified QoS port by using the **qos port maximum egress-bandwidth** command.

The maximum egress bandwidth value may be entered as an integer, in bits-per-second or with abbreviated units (for example, 10k, 5m, 1g, 1t). If the maximum egress bandwidth value is specified as an integer, without an abbreviated unit designation, the value is applied in kbps by default.

For example, if the number 10 is specified, 10K is the maximum egress bandwidth value used. However, if 10G is specified, the maximum egress bandwidth value applied is 10 gbps.

```
-> qos port 3/1 maximum egress-bandwidth 10
-> qos port 4/1-8 maximum egress-bandwidth 10g
```

Configuring Maximum Ingress Bandwidth

Configure the maximum rate at which traffic is received on a QoS port by using the **qos port maximum ingress-bandwidth** command.

The maximum ingress bandwidth value may be entered as an integer, in bits-per-second or with abbreviated units (for example, 10k, 5m, 1g, 1t). If the maximum ingress bandwidth value is specified as an integer, without an abbreviated unit designation, the value is applied in kbps by default.

For example, if the number 10 is specified, 10K is the maximum ingress bandwidth value used. However, if 10G is specified, the maximum ingress bandwidth value applied is 10 gbps.

```
-> qos port 3/1 maximum ingress-bandwidth 10
-> qos port 4/1-8 maximum ingress-bandwidth 10g
```

Configuring Maximum Queue Depth

Configure the maximum queue depth or bucket size assigned to this action, in bytes, used for traffic metering. The queue depth or bucket size determines the amount of buffer allocated to each queue. When the queue depth or bucket size is reached, the switch starts dropping packets.

The maximum ingress or egress queue depth can be configured using the **qos port maximum depth** command.

The maximum bucket size value may be entered as an integer, in bytes or with abbreviated units (for example, 10k, 5m). If the maximum depth value is specified as an integer, without an abbreviated unit designation, the value is applied in Kbytes by default.

For example, if the number 10 is specified as the maximum ingress depth, 10K is the maximum depth value used. However, if 1M is specified as the maximum egress depth, the maximum egress depth value applied is 1 Mbyte.

```
-> qos port 3/1 maximum ingress-depth 10
-> qos port 4/1-8 maximum egress-depth 1m
```

QoS Policy Overview

A policy (or a *policy rule*) is made up of a condition and an action. The condition specifies parameters that the switch examines in incoming flows, such as destination address or Type of Service (ToS) bits. The action specifies what the switch does with a flow that matches the condition; for example, it can queue the flow with a higher priority, or reset the ToS bits.

Policies can be created directly on the switch through the CLI or WebView or policies can be created on an external LDAP server through the PolicyView application. The switch makes a distinction between policies created on the switch and policies created on an LDAP server.

Note. Polices can only be modified using the same source used to create them. Policies configured through PolicyView can only be edited through PolicyView. Policies created directly on the switch through the CLI or WebView can only be edited on the switch. Policies are created through the CLI or WebView, however, to override policies created in PolicyView. And vice versa.

This section discusses policy configuration using the CLI. For information about using WebView to configure the switch, see the *OmniSwitch AOS Release 8 Switch Management Guide*. For information about configuring policies through PolicyView, see the PolicyView online help.

How Policies Are Used

When a flow comes into the switch, the QoS software in the switch checks to see if there are any policies with conditions that match the flow.

- ***If there are no policies that match the flow***, the flow is accepted and the default QoS port settings for priority are used to classify and mark the flow.
- ***If there is more than one policy that matches the flow***, the policy with the highest precedence is applied to the flow. For more information about policy precedence, see "Rule Precedence" on page 28-49.
- ***Flows must also match all parameters configured in a policy condition.*** A policy condition must have at least one classification parameter.

Once the flow is classified and matched to a policy, the switch enforces the policy by mapping each packet of the flow to the appropriate queue and scheduling it on the output port. There are a total of eight queues per port. Traffic is mapped to a queue based on policies, the ToS/802.1p value of the packet, and whether the port is trusted or untrusted. For more information about queues, see "Congestion Management" on page 28-11.

Policy Lists

A QoS policy list provides a method for grouping multiple policy rules together and applying the group of rules to specific types of traffic.The type of traffic to which a policy list is applied is determined by the type of list that is configured. There are four types of policy lists:

- **Default**—All rules are associated with a default policy list when the rules are created. This list is not configurable, but it is possible to direct QoS to not assign a rule to this list.
- **User Network Profile (UNP)**—This type of policy list is associated with a Universal Network Profile. The rules in this list are applied to device traffic that is classified into the profile.
- **Egress**—The rules in this type of policy list are applied to traffic egressing on switch ports.
- **Application Fingerprinting (AFP)**—The rules in this type of policy list are applied to device traffic received on Application Fingerprinting interfaces. *AFP is supported only on the OmniSwitch 6900.*

For more information, see "Creating Policy Lists" on page 28-50.

Interaction With Other Features

QoS policies are an integral part of configuring other switch features, such as Link Aggregation. In addition, QoS settings can affect other features in the switch; or QoS settings can require that other switch features be configured in a particular way.

A summary of related features is given here:

- **Dynamic Link Aggregates**—Policies can be used to prioritize dynamic link aggregation groups. For details, see Chapter 10, "Configuring Dynamic Link Aggregation."
- **802.1Q**—Tagged ports are always untrusted by default. For information about configuring ports with 802.1Q, see Chapter 4, "Configuring VLANs."
- **LDAP Policy Management**—Policies can also be configured through the PolicyView application and stored on an attached LDAP server. LDAP policies can only be modified through PolicyView. For information about setting up a policy server and managing LDAP policies, see Chapter 29, "Managing Policy Servers."
- **VLAN Stacking Ethernet Service**—VLAN Stacking ports are always trusted and the default classification is set to 802.1p. QoS policy conditions to match the inner VLAN tag and inner 802.1p tag are available for classifying customer information contained in VLAN Stacking frames. For information about VLAN Stacking see Chapter 37, "Configuring VLAN Stacking."
- **Universal Network Profiles (UNP)**—The UNP feature provides the ability to assign a list of QoS policy rules to a profile. The rules contained in the list are applied to any device that is assigned to the UNP. For more information about policy lists, see "Policy Lists" on page 28-30 and Chapter 30, "Configuring Access Guardian."

Valid Policies

The switch does not allow you to create invalid condition/action combinations; if you enter an invalid combination, an error message is displayed. A list of valid condition and actions is given in "Policy Conditions" on page 28-31 and "Policy Actions" on page 28-32.

It is possible to configure a valid QoS rule that is active on the switch, however the switch is not able to enforce the rule because some other switch function (for example, routing) is disabled.

Policy Conditions

The following conditions are supported and can be combined with other conditions and/or actions:

Supported Policy Conditions Table

Layer 1	Layer 2	Layer 3
destination port destination port group source port source port group	source MAC source MAC group destination MAC destination MAC group 802.1p inner 802.1p ethertype source VLAN inner source VLAN destination VLAN (multicast rules only)	IP protocol source IP multicast IP destination IP source network group destination network group multicast network group ToS, DSCP ICMP type, ICMP code source IPv6 destination IPv6 IPv6 traffic IPv6 flow label (FL)
Layer 4	**IP Multicast (IGMP)**	
source TCP/UDP port destination TCP/UDP port service, service group TCP flags (ECN/CWR are not supported)	destination only	

The CLI prevents you from configuring invalid condition combinations that are never allowed; however, it does allow you to create combinations that are supported in some scenarios. For example, you might configure **source ip** and a **destination ip** for the same condition.

Consider the following guidelines when configuring policy conditions:

- IPv4 and IPv6 conditions cannot be combined.
- The destination VLAN condition is only supported in multicast policy rules.
- IP multicast traffic (not IGMP) is treated as regular traffic; QoS functionality works the same way with this type of traffic, with the exception that the destination port condition does not apply.
- The IP multicast condition works in combination with Layer 1, Layer 2, and Layer 3 destination conditions only if these conditions specify the device that sends the IGMP report packet.
- Source and destination parameters can be combined in Layer 2, Layer 3, and Layer 4 conditions.
- In a given rule, ToS or DSCP can be specified for a condition with priority specified for the action.
- Individual items and their corresponding groups cannot be combined in the same condition. For example, a source IP address cannot be included in a condition with a source IP network group.
- The Layer 1 destination port condition only applies to bridged traffic, not routed traffic. In addition, the destination port condition only applies to unicast bridged traffic, not multicast or broadcast traffic.
- Layer 2 and Layer 3 rules are always effected on bridged and routed traffic. As a result, combining source or destination TCP/UDP port and IP protocol in a condition is allowed.

For specific information about how to configure policy conditions and actions to create a policy rule, see "Creating Policies" on page 28-43.

Policy Actions

The following actions are supported and can be combined with other actions.

Supported Policy Actions Table

- ACL (disposition accept, drop, deny)
- Priority/CoS
- 802.1p ToS/DCSP Stamping and Mapping (only applies to the outer 802.1p value; cannot modify the inner value)
- Maximum Bandwidth
- Maximum Depth
- Tri-Color Marking (TCM) Rate Limiting
- Shared (shares the bandwidth rate between rules that specify the same maximum bandwidth action)
- Port Redirection
- Link Aggregate Redirection
- No Cache (disables the logging of rule entries to the
- hardware cache)
- Port Disable
- Permanent Gateway IPv4/IPv6
- Mirror

The CLI prevents you from configuring invalid action combinations that are never allowed; however, it does allow you to create combinations that are supported in some scenarios. For example, an action specifying maximum bandwidth can be combined with an action specifying priority.

Use the following "Policy Action Combinations Table" together with the "Supported Policy Actions Table" as a guide when creating policy actions.

Policy Action Combinations Table

	Drop	Priority	Stamp/ Map	Max BW	Redirect Port	Redirect Linkagg	Port Disable	Permanent Gateway IP	Mirror
Drop	N/A	No	No	No	No	No	No	No	Yes
Priority	No	N/A	Yes	Yes	Yes	Yes	No	Yes	Yes
Stamp/Map	No	Yes	N/A	Yes	Yes	Yes	No	Yes	Yes
Max BW	No	Yes	Yes	N/A	Yes	Yes	No	Yes	Yes
Redirect Port	No	Yes	Yes	Yes	N/A	No	No	Yes	Yes
Redirect Linkagg	No	Yes	Yes	Yes	No	N/A	No	Yes	Yes
Port Disable	No	No	No	No	No	No	N/A	No	No
Permanent Gateway IP	No	Yes	Yes	Yes	Yes	Yes	No	N/A	Yes
Mirroring	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	N/A

For specific information about how to configure policy conditions and actions to create a policy rule, see "Creating Policies" on page 28-43.

Condition and Action Combinations

Conditions and actions are combined in policy rules. The CLI prevents you from configuring invalid condition/action combinations that are never allowed; however, the following table provides a quick reference for determining which condition/action combinations are *not* valid. Each row represents a policy condition or conditions combined with the policy action or actions in the same row.

Policy Condition/Action Combinations

Conditions	Actions	Supported When?
multicast IP address *or* network group	all actions	never, except with disposition action
multicast IPv6 address	all actions	never, except with disposition and mirror actions
destination VLAN	all actions	never, except with disposition action in a multicast rule (a rule that uses the "multicast" keyword and only applies to IGMP traffic)
destination slot/port or port group	all actions	bridging only

QoS Defaults

The following tables list the defaults for global QoS parameters, individual port settings, policy rules, default policy rules, and queue management profiles.

Global QoS Defaults

Use the **qos reset** command is to reset global values to their defaults.

Description	Command	Default
QoS enabled or disabled	**qos**	enabled
Whether ports are globally trusted or untrusted	**qos trust-ports**	VLAN Stacking ports are always trusted; all other port types are untrusted
Statistics interval	**qos stats interval**	60 seconds
Level of log detail	**qos log level**	5
Number of lines in QoS log	**qos log lines**	10000
Whether log messages are sent to the console	**qos log console**	no
Whether log messages are available to OmniVista applications	**qos forward log**	no
Whether IP anti-spoofing is enabled on UserPorts.	**qos user-port filter**	yes
Whether a UserPorts port is administratively disabled when unwanted traffic is received.	**qos user-port shutdown**	no
Global default DEI bit setting for ports	**qos dei**	disabled
Priority for IP Phone connections.	**qos phones**	trusted

QoS Port Defaults

Use the **qos port reset** command to reset port settings to the defaults.

Description	Command/keyword	Default
Whether the port is trusted or untrusted	**qos port trusted**	VLAN Stacking ports are always trusted; all other port types are untrusted.
The maximum egress bandwidth	**qos port maximum egress-bandwidth**	port bandwidth
The maximum ingress bandwidth	**qos port maximum ingress-bandwidth**	port bandwidth
The maximum ingress or egress queue depth or bucket size	**qos port maximum depth**	0

Description	Command/keyword	Default
The default 802.1p value inserted into packets received on untrusted ports.	**qos port default 802.1p**	0
The default DSCP value inserted into packets received on untrusted ports.	**qos port default dscp**	0
The default egress classification value inserted into packets received on trusted ports.	**qos port default classification**	DSCP (802.1p for VLAN Stacking ports).
The Drop Eligible Indicator (DEI) bit setting.	**qos port dei**	disabled

Queue Management Defaults

The queue management and related QoS functions are implemented using a Queue Set (QSet) framework. Each port and link aggregate is associated with a set of eight egress queues, referred to as a Queue Set Instance (QSI). Each QSI is associated with QSet profile 1 (QSP 1) by default.

A QSP defines both global parameters for the profile and individual queue profile parameters that are applied to the eight queues associated with the QSet instance. See "Congestion Management" on page 28-11 for more information.

The following are the default QSet Instance (QSI) settings applied to each port or link aggregate:

Port QSI	Default
QSet Profile	QSP 1
Statistics Admin Status	Disabled
Statistics Interval	60 seconds
Bandwidth	100%

The following are the default QSet Profile (QSP 1) settings applied to each QSI:

QSP 1	Default
Bandwidth	100%
QP1–QP8 Queue Type	Strict Priority
QP1–QP8 CIR PIR	0%, 100%
WFQ Mode	WERR
WFQ Weight	0

Policy Rule Defaults

The following are defaults for the **policy rule** command:

Description	Keyword	Default
Policy rule enabled or disabled	**enable | disable**	enabled
Determines the order in which rules are searched	**precedence**	0
Whether the rule is saved to flash immediately	**save**	yes
Whether messages about flows that match the rule are logged	**log**	no
How often to check for matching flow messages	**log interval**	60 seconds
Whether to count bytes or packets that match the rule.	**count**	packets
Whether to send a trap for the rule.	**trap**	yes (trap sent only on port disable action or UserPort shutdown operation)
Whether the rule is saved to the default list	**default-list**	yes (all policy rules belong to the default list unless otherwise specified at the time the rule is created)

Policy Action Defaults

The following are defaults for the **policy action** command:

Description	Keyword	Default
Whether the flow matching the rule must be accepted or denied	**disposition**	accept
Tri-Color Marking (TCM) mode - committed rate and burst size - peak rate and burst size	 **cir cbs** **pir pbs**	Single-rate TCM (srTCM) mode CIR=0, CBS=0 PIR=0, PBS=0

Note that in the current software release, the **deny** and **drop** options produce the same effect that is, the traffic is silently dropped.

Note. There are no defaults for the **policy condition** command.

Default (Built-in) Policies

The switch includes some built-in policies, or default policies, for particular traffic types or situations where traffic does not match any policies. In all cases, the switch accepts the traffic and places it into default queues.

- *Other traffic*—Any traffic that does not match a policy is accepted.
- *The* **switch** *and* **switch6** *network group*—The switch has two default network groups:
  - The **switch** network group includes all of the IPv4 addresses configured for the switch.
  - The **switch6** network group includes all of the IPv6 addresses configured for the switch.

Note: IP interfaces associated with services are not included in the default network groups.

Both default network groups can be used in policies. See "Creating Network Groups" on page 28-55 for more information about network groups.

Configuring QoS

QoS configuration involves the following general steps:

1 Configuring Global Parameters. In addition to enabling/disabling QoS, global configuration includes settings such as global port parameters and various timeouts. The type of parameters you might want to configure globally depends on the types of policies you can configure. For example, if you want to set up policies for 802.1p or ToS/DSCP traffic, you can configure all ports as trusted ports.

Typically, you need not change any of the global defaults. See "Global QoS Defaults" on page 28-34 for a list of the global defaults. See "Configuring Global QoS Parameters" on page 28-39 for information about configuring global parameters.

2 Configuring QoS Port Parameters. This configuration includes setting up QoS parameters on a per port basis. Typically you do not need to change the port defaults. See "QoS Port Defaults" on page 28-34 for a list of port defaults. See "Classification" on page 28-5 and "Traffic Policing and Shaping" on page 28-21 for information about configuring port parameters.

3 Configuring Queue Set (QSet) Profiles. The queue management configuration is applied using embedded QSet profiles. A default profile configuration is applied when the switch comes up. Selecting different profiles is only necessary if the default profile settings are not sufficient. See "Queue Management Defaults" on page 28-35 for a list of default profile settings. See "Congestion Management" on page 28-11 for information about configuring QSet profiles.

4 Setting Up Policies. Most QoS configuration involves setting up policies. In addition, policy lists are configurable for use with the Universal Network Profile (UNP) feature. See "Creating Policies" on page 28-43.

5 Applying the Configuration. All policy rule configuration and some global parameters must be specifically applied through the **qos apply** command before they are active on the switch. See "Applying the Configuration" on page 28-72.

Configuring Global QoS Parameters

This section describes the global QoS configuration, which includes enabling and disabling QoS, applying and activating the configuration, controlling the QoS log display, and configuring QoS port and queue parameters.

Enabling/Disabling QoS

By default QoS is enabled on the switch. If QoS policies are configured and applied, the switch attempts to classify traffic and apply relevant policy actions.

To disable the QoS, use the **qos** command. For example:

```
-> qos disable
```

QoS is immediately disabled. When QoS is disabled globally, any flows coming into the switch are not classified (matched to policies).

To re-enable QoS, enter the **qos** command with the **enable** option:

```
-> qos enable
```

QoS is immediately re-enabled. Any policies that are active on the switch are used to classify traffic coming into the switch.

Note that individual policy rules can be enabled or disabled with the **policy rule** command.

Using the QoS Log

The QoS software in the switch creates its own log for QoS-specific events. You can modify the number of lines in the log or change the level of detail given in the log. The PolicyView application, which is used to create QoS policies stored on an LDAP server, query the switch for log events; or log events can be immediately available to the PolicyView application through a CLI command. Log events can also be forwarded to the console in real time.

What Kind of Information Is Logged

The **debug qos** command controls what kind of information is displayed in the log. The **qos log level** command determines how specific the log messages are. See "Log Detail Level" on page 28-40.

By default, only the most basic QoS information is logged. The types of information that can be logged includes rules, Layer 2 and Layer 3 information, etc. For a detailed explanation about the types of information that can be logged, see the **debug qos** command page in the *OmniSwitch AOS Release 8 CLI Reference Guide*. A brief summary of the available keywords is given here:

debug qos keywords		
info	**sl**	**classifier**
config	**mem**	**sem**
rule	**mapper**	**pm**
main	**slot**	**ingress**
port	**l2**	**egress**
msg	**l3**	

To display information about any QoS rules on the switch, enter **debug qos rule**:

```
-> debug qos rule
```

To change the type of debugging, use **no** with the relevant type of information that you want to remove. For example:

```
-> debug qos no rule
```

To turn off debugging (which effectively turns off logging), enter the following:

```
-> no debug qos
```

Enter the **qos apply** command to activate the setting.

Number of Lines in the QoS Log

By default the QoS log displays a maximum of 10000 lines. To change the maximum number of lines that can display, use the **qos log lines** command and enter the number of lines. For example:

```
-> qos log lines 30
```

The number of lines in the log is changed. To activate the change, enter the **qos apply** command.

Note. If you change the number of log lines, the QoS log can be completely cleared. To change the log lines without clearing the log, set the log lines in the **boot.cfg** file; the log is set to the specified number of lines at the next reboot.

Log Detail Level

To change the level of detail in the QoS log, use the **qos log level** command. The log level determines the amount of detail that is given in the QoS log. The **qos log level** command is associated with the **qos debug** command, which determines what kind of information is included in the log.

The default log level is 5. The range of values is 1 (lowest level of detail) to 8 (highest level of detail). For example:

```
-> qos log level 7
```

The log level is changed immediately but the setting is not saved in flash. To activate the change, enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 28-72.

Note. A high log level value impacts the performance of the switch.

Forwarding Log Events

NMS applications query the switch for logged QoS events. Use the **qos forward log** command to make QoS log events available to these applications in real time. For example:

```
-> qos forward log
```

To disable log forwarding, enter the following command:

```
-> qos no forward log
```

To activate the change, enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 28-72.

If event forwarding is disabled, NMS applications can still query the QoS software for events, but the events are not sent in real time.

Forwarding Log Events to the Console

QoS log messages can be sent to the switch logging utility, which is an event logging application available on the OmniSwitch. The configuration of the switch logging utility then determines if QoS messages are sent to a log file in the switch's flash file system, displayed on the switch console, and/or sent to a remote syslog server.

To send log events to the switch logging utility, enter the following command:

```
-> qos log console
```

To disable immediate forwarding of events to switch logging, enter the following command:

```
-> qos no log console
```

To activate the change, enter the **qos apply** command. For more information about the **qos apply** command, see "Applying the Configuration" on page 28-72.

Use the **swlog output** command to configure switch logging to output logging events to the console. Note that this is in addition to sending log events to a file in the flash file system of the switch. See the "Using Switch Logging" chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide* for more information.

Displaying the QoS Log

To view the QoS log, use the **show qos log** command. The display is similar to the following:

```
**QOS Log**

Insert rule 0
Rule index at 0
Insert rule 1
Rule index at 1
Insert rule 2
Rule index at 2
Enable rule r1 (1) 1,1
Enable rule r2 (0) 1,1
Enable rule yuba1 (2) 1,1
Verify rule r1(1)
Enable rule r1 (1) 1,1
Really enable r1
Update condition c1 for rule 1 (1)
Verify rule r2(1)
Enable rule r2 (0) 1,1
Really enable r2
Update condition c2 for rule 0 (1)
Verify rule yuba1(1)
Enable rule yuba1 (2) 1,1
Really enable yuba1
Update condition yubamac for rule 2 (1)
QoS Manager started TUE MAR 10 13:46:50 2002

Match rule 2 to 1
Match rule 2 to 2
Match rule 2 to 3
```

The log display can be modified through the **qos log lines**, **qos log level**, and **debug qos** commands. The log display can also be output to the console through the **qos log console** command or sent to the policy

software in the switch (which manages policies downloaded from an LDAP server) through the **qos forward log** command.

Clearing the QoS Log

The QoS log can get large if invalid rules are configured on the switch, or if a lot of QoS events have taken place. Clearing the log makes the file easier to manage.

To clear the QoS log, use the **clear qos log** command. For example:

```
-> clear qos log
```

All the current lines in the QoS log are deleted.

Setting the Statistics Interval

To change how often the switch polls the network interfaces for QoS statistics, use the **qos stats interval** command with the desired interval time in seconds. The default is 60 seconds. For example:

```
-> qos stats interval 30
```

Statistics are displayed through the **show qos statistics** command. For more information about this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Returning the Global Configuration to Defaults

To return the global QoS configuration to its default settings, use the **qos reset** command. The defaults then become active on the switch. For a list of global defaults, see "QoS Defaults" on page 28-34.

Note. The **qos reset** command only affects the global configuration. It does not affect any policy configuration.

Verifying Global Settings

To display information about the global configuration, use the following **show** commands:

show qos config	Displays global information about the QoS configuration.
show qos statistics	Displays statistics about QoS events.

For more information about the syntax and displays of these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Creating Policies

This section describes how to create policies in general. For information about configuring specific types of policies, see "Policy Applications" on page 28-75.

Basic commands for creating policies are as follows:

policy condition
policy action
policy rule

This section describes generally how to use these commands. For additional details about command syntax, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Note. A policy rule can include a policy condition or a policy action that was created through PolicyView rather than the CLI. But a policy rule, policy action, or policy condition can only be modified through the source that created it. For example, if an action was created in PolicyView, it can be included in a policy rule configured through the CLI, but it cannot be modified through the CLI.

Policies are not used to classify traffic until the **qos apply** command is entered. See "Applying the Configuration" on page 28-72.

Quick Steps for Creating Policies

Follow the steps below for a quick tutorial on creating policies. More information about how to configure each command is given in later sections of this chapter.

1. Create a policy condition with the **policy condition** command. For example:

```
-> policy condition cond3 source ip 10.10.2.3
```

2. Create a policy action with the **policy action** command. For example:

```
-> policy action action2 priority 7
```

3. Create a policy rule with the **policy rule** command. For example:

```
-> policy rule my_rule condition cond3 action action2
```

4. Use the **qos apply** command to apply the policy to the configuration. For example:

```
-> qos apply
```

Note. (*Optional*) To verify that the rule has been configured, use the **show policy rule** command. The display is similar to the following:

```
 -> show policy rule
Rule name                        : my_rule
  Condition name                 = cond3,
  Action name                    = action2,
```

An example of how the example configuration commands might display when entered sequentially on the command line is given here:

```
-> policy condition cond3 source ip 10.10.2.3
-> policy action action2 priority 7
-> policy rule my_rule condition cond3 action action2
-> qos apply
```

ASCII-File-Only Syntax

When the **policy rule**, **policy condition**, and **policy action** commands as well as any of the condition group commands are configured and saved in an ASCII file (typically through the **snapshot** command), the commands included in the file include syntax indicating the origin of the command. The origin specifies where the rule, condition, condition group, or action was created, either an LDAP server or the CLI (**from ldap** or **from cli**). For built-in QoS objects, the syntax displays as **from blt**. For example:

```
-> policy action A2 from ldap disposition accept
```

The **from** option is configurable (for LDAP or CLI only) on the command line; however, it is not recommended that a QoS object's origin be modified. The **blt** keyword indicates built-in; this keyword cannot be used on the command line. For information about built-in policies and QoS groups, see "How Policies Are Used" on page 28-29.

Creating Policy Conditions

This section describes how to create policy conditions in general. Creating policy conditions for particular types of network situations is described later in this chapter.

Note. Policy condition configuration is not active until the **qos apply** command is entered. See "Applying the Configuration" on page 28-72.

To create or modify a policy condition, use the **policy condition** command with the keyword for the type of traffic you want to classify, for example, an IP address or group of IP addresses. In this example, a condition (**c3**) is created for classifying traffic from source IP address 10.10.2.1:

```
-> policy condition c3 source ip 10.10.2.1
```

There are many options for configuring a condition, depending on how you want the switch to classify traffic for this policy. An overview of the options is given here. Later sections of this chapter describe how to use the options in particular network situations.

Note. The group options in this command refer to groups of addresses, services, or ports that you configure separately through policy group commands. Rather than create a separate condition for each address, service, or port, use groups and attach the group to a single condition. See "Using Condition Groups in Policies" on page 28-54 for more information about setting up groups.

More than one condition parameter can be specified. Some condition parameters are mutually exclusive. For supported combinations of condition parameters, see "Policy Conditions" on page 28-31.

policy condition keywords		
source ip	**service**	**source vlan**
source ipv6	**service group**	**inner source vlan**
destination ip		**destination vlan (multicast only)**
destination ipv6	**icmptype**	
multicast ip	**icmpcode**	**source port**
multicast ipv6	**ip-protocol**	**source port group**
source network group		**destination port**
destination network group	**ipv6**	**destination port group**
multicast network group	**flow-label**	
source ip-port		**vrf**
destination ip-port	**tos**	**fragments**
source tcp-port	**dscp**	
destination tcp-port	**802.1p**	**app-mon-application-group**
source udp-port	**inner 802.1p**	**app-mon-application-name**
destination udp-port		**appfp-group**
	source mac	**sip**
ethertype	**destination mac**	
established	**source mac group**	
tcpflags	**destination mac group**	

The condition is not activated on the switch until you enter the **qos apply** command.

Removing Condition Parameters

To remove a classification parameter from the condition, use **no** with the relevant keyword. For example:

```
-> policy condition c3 no source ip
```

The specified parameter (in this case, a source IP address) is removed from the condition (**c3**) at the next **qos apply**.

Note. You cannot remove all parameters from a policy condition. A condition must be configured with at least one parameter.

Deleting Policy Conditions

To remove a policy condition, use the **no** form of the command. For example:

```
-> no policy condition c3
```

The condition (**c3**) cannot be deleted if it is currently being used by a policy rule. If a rule is using the condition, the switch displays an error message. For example:

```
ERROR: c3 is being used by rule 'my_rule'
```

In this case, the condition is not deleted. The condition (**c3**) must first be removed from the policy rule (**my_rule**). See "Creating Policy Rules" on page 28-47 for more information about setting up rules.

If **c3** is not used by a policy rule, it is deleted after the next **qos apply**.

Creating Policy Actions

This section describes how to configure policy actions in general. Creating policy actions for particular types of network situations is described later in this chapter.

To create or modify a policy action, use the **policy action** command with the desired action parameter. A policy action must specify the way traffic must be treated. For example, it might specify a priority for the flow, a source address to rewrite in the IP header, or it can specify that the flow is dropped. For example:

```
-> policy action Block disposition drop
```

In this example, the action (**Block**) has a disposition of **drop** (disposition determines whether a flow is allowed or dropped on the switch). This action can be used in a policy rule to deny a particular type of traffic specified by a policy condition.

Note. Policy action configuration is not active until the **qos apply** command is entered. See "Applying the Configuration" on page 28-72.

More than one action parameter can be specified. Some parameters are mutually exclusive. In addition, some action parameters are only supported with particular condition parameters. For information about supported combinations of condition and action parameters, see "Policy Conditions" on page 28-31 and "Policy Actions" on page 28-32. See the *OmniSwitch AOS Release 8 CLI Reference Guide* for details about command syntax.

policy action keywords	
disposition	**dcsp**
shared	**map**
priority	**permanent gateway-ip**
maximum bandwidth	**permanent gateway-ipv6**
maximum depth	**port-disable**
cir cbs pir pbs color-only	**redirect port**
cpu priority	**redirect linkagg**
tos	**no-cache**
802.1p	**mirror**

Note. If you combine **priority** with **802.1p**, **dscp**, **tos**, or **map**, in an action, the priority value is used to prioritize the flow.

Removing Action Parameters

To remove an action parameter or return the parameter to its default, use **no** with the relevant keyword.

```
-> policy action a6 no priority
```

This example removes the configured priority value from action **a6**. If any policy rule is using action **a6**, the default action is to allow the flow classified by the policy condition.

The specified parameter (in this case, priority) is removed from the action at the next **qos apply**.

Deleting a Policy Action

To remove a policy action, use the **no** form of the command.

```
-> no policy action a6
```

The action cannot be deleted if it is currently being used by a policy rule. If a rule is using the action, the switch displays an error message. For example:

```
ERROR: a6 is being used by rule 'my_rule'
```

In this case, the action is not deleted. The action (**a6**) must first be removed from the policy rule (**my_rule**). See "Creating Policy Rules" on page 28-47 for more information about setting up rules.

If **a6** is not used by a policy rule, it is deleted after the next **qos apply**.

Creating Policy Rules

This section describes in general how to create or delete policy rules and rule parameters. See later sections of this chapter for more information about creating particular types of policy rules.

To create a policy rule, use the **policy rule** command and specify the name of the rule, the desired condition, and the desired action.

In this example, condition **c3** is created for traffic coming from IP address 10.10.8.9, and action **a7** is created to prioritize the flow. Policy rule **rule5** combines the condition and the action, so that traffic arriving on the switch from 10.10.8.9 is placed into the highest priority queue.

```
-> policy condition c3 source ip 10.10.8.9
-> policy action a7 priority 7
-> policy rule rule5 condition c3 action a7
```

The rule (**rule5**) only takes effect after the **qos apply** command is entered. For more information about the **qos apply** command, see "Applying the Configuration" on page 28-72.

The **policy rule** command can specify the following keywords:

policy rule keywords
precedence
validity period
save
log
log interval
count
trap

In addition, a policy rule can be administratively disabled or re-enabled using the **policy rule** command. By default rules are enabled. For a list of rule defaults, see "Policy Rule Defaults" on page 28-36.

Information about using the **policy rule** command options is given in the next sections.

Configuring a Rule Validity Period

A validity period specifies the days and times during which a rule is in effect. By default there is no validity period associated with a rule, which means the rule is always active.

To configure the days, months, times, and/or time intervals during which a rule is active, use the **iec message-type priority** command. Once the validity period is defined, it is then associated with a rule using the **policy rule** command. For example, the following commands create a validity period named **vp01** and associate it with rule **r01**:

```
-> policy validity period vp01 hours 13:00 to 19:00 days monday friday
-> policy rule r01 validity period vp01
```

Note the following when using validity periods to restrict the times when a rule is active:

- Only one validity period is associated with a policy rule. Each time this command is entered with a validity period name specified, the existing period name is overwritten with the new one.
- A rule is only in effect when all the parameters of its validity period are true. In the above example, rule **r01** is only applied between 13:00 and 19:00 on Mondays and Fridays. During all other times and days, the rule is not applied.
- Software and hardware resources are allocated for rules associated with a validity period even if the validity period is not active. Pre-allocating the resources makes sure the rule can be enforced when the validity period becomes active.

Disabling Rules

By default, rules are enabled. Rules are disabled or re-enabled through the **policy rule** command using the **disable** and **enable** options. For example:

```
-> policy rule rule5 disable
```

This command prevents **rule5** from being used to classify traffic.

Note. If **qos disable** is entered, the rule is not used to classify traffic even if the rule is enabled. For more information about enabling/disabling QoS globally, see "Enabling/Disabling QoS" on page 28-39.

Rule Precedence

The switch attempts to classify flows coming into the switch according to policy precedence. Only the rule with the highest precedence is applied to the flow. This is true even if the flow matches more than one rule.

Precedence is particularly important for Access Control Lists (ACLs). For more details about precedence and examples for using precedence, see Chapter 28, "Configuring QoS."

How Precedence is Determined

When there is a conflict between rules, precedence is determined using one of the following methods:

- **Precedence value**—Each policy has a precedence value. The value is user-configured through the **policy rule** command in the range from 0 (lowest) to 65535 (highest). (The range 30000 to 65535 is typically reserved for PolicyView.) By default, a policy rule has a precedence of 0.
- **Configured rule order**—If a flow matches more than one rule and both rules have the same precedence value, the rule that was *configured first* in the list takes precedence.

Specifying Precedence for a Particular Rule

To specify a precedence value for a particular rule, use the **policy rule** command with the precedence keyword. For example:

```
-> policy rule r1 precedence 200 condition c1 action a1
```

Saving Rules

The **save** option marks the policy rule so that the rule is captured in an ASCII text file (using the **configuration snapshot** command) and saved to the working directory (using the **write memory** command). By default, rules are saved.

If the **save** option is removed from a rule, the **qos apply** command activates the rule for the current session, but the rule is not saved over a reboot. Typically, the **no save** option is used for temporary policies that you do not want saved in the switch configuration file.

To remove the **save** option from a policy rule, use **no** with the **save** keyword. For example:

```
-> policy rule rule5 no save
```

To reconfigure the rule as saved, use the **policy rule** command with the **save** option. For example:

```
-> policy rule rule5 save
```

For more information about the **configuration snapshot**, **write memory**, and **copy running-config working** commands, see the *OmniSwitch AOS Release 8 Switch Management Guide* and the *OmniSwitch AOS Release 8 CLI Reference Guide.*

For more information about applying rules, see "Applying the Configuration" on page 28-72.

Logging Rules

Logging a rule is useful for determining the source of firewall attacks. To specify that the switch must log information about flows that match the specified policy rule, use the **policy rule** command with the **log** option. For example:

```
-> policy rule rule5 log
```

To stop the switch from logging information about flows that match a particular rule, use **no** with the **log** keyword. For example:

```
-> policy rule rule5 no log
```

When logging is active for a policy rule, a logging interval is applied to specify how often to look for flows that match the policy rule. By default, the interval time is set to 30 seconds. To change the log interval time, use the optional **interval** keyword with the log option. For example:

```
-> policy rule rule5 log interval 1500
```

Note that setting the log interval time to 0 specifies to log as often as possible.

Deleting Rules

To remove a policy rule, use the **no** form of the command.

```
-> no policy rule rule1
```

The rule is deleted after the next **qos apply**.

Creating Policy Lists

A QoS policy list provides a method for grouping multiple policy rules together and applying the group of rules to specific types of traffic.The type of traffic to which a policy list is applied is determined by the type of list that is configured. There are four types of policy lists:

- **Default**—This list is always available on every switch and is not configurable. By default, a policy rule is associated with this list when the rule is created. All default list rules are applied to ingress traffic.
- **Universal Network Profile (UNP)**—This type of policy list is associated with a UNP. The rules in this list are applied to ingress traffic that is classified by the UNP. See Chapter 30, "Configuring Access Guardian,"for more information.
- **Egress**—The rules in this type of policy list are applied to traffic egressing on switch ports.
- **Application Fingerprinting (AFP)**—The rules in this type of configurable policy list are applied to device traffic received on Application Fingerprinting interfaces. See Chapter 32, "Configuring Application Fingerprinting,"for more information. *AFP is supported only on the OmniSwitch 6900.*

To create a UNP policy list, use the **policy list** command to specify a list name and type and then use the **policy list rules** command to specify the names of one or more existing QoS policy rules to add to the list. For example, the following commands create two policy rules and associates these rules with the **unp1_rules** list:

```
-> policy condition c1 802.1p 5
-> policy action a1 disposition drop
-> policy rule r1 condition c1 action a1 no default-list
-> policy condition c2 source ip 10.5.5.0
-> policy action a2 disposition accept
-> policy rule r2 condition c2 action a2 no default-list
-> policy list unp1_rules type unp enable
-> policy list unp1_rules rules r1 r2
-> qos apply
```

Note that the **no default-list** option was used to create the rules. Using this option is recommended when creating a policy list for a UNP. See "Guidelines for Configuring Policy Lists" on page 28-51.

The following example creates a policy rule (**rule1**) that is automatically assigned to the default policy list.

```
-> policy condition cond1 source mac 00:11:22:33:44:55 source vlan 100
-> policy action act1 disposition drop
-> policy rule rule1 condition cond1 action act1
-> qos apply
```

In this example, the **no default-list** parameter is *not* used with the **policy rule** command, so the rule is automatically assigned to the default policy list. The default list always exists and is not configurable. As a result, the **policy list** command is not required to assign the rule to the default list.

By default, a policy list is enabled at the time the list is created. To disable or enable a policy list, use the following commands:

```
-> policy list unp1_rules disable
-> policy list unp1_rules enable
```

To remove an individual rule from a policy list, use the following command:

```
-> policy list unp1_rules no rules r2
```

To remove an entire policy list from the switch configuration, use the following command:

```
-> no policy list unp1_rules
```

Use the **show policy list** command to display the QoS policy rule configuration for the switch.

Guidelines for Configuring Policy Lists

Consider the following guidelines when configuring QoS policy rules and lists:

- Create policy rules first before attempting to create a list. The **policy list rules** command requires that the specified policy rules must already exist in the switch configuration. See "Creating Policies" on page 28-43.
- If a rule is a member of multiple policy lists but one or more of these lists are disabled, the rule is still active for those lists that are enabled.
- If the QoS status of an individual rule is disabled, then the rule is disabled for all policy lists, even if a list to which the policy belongs is enabled.
- By default, QoS assigns rules to the default policy list. To exclude a rule from this list, use the **no default-list** option of the **policy rule** command when the rule is created. See "Using the Default Policy List" on page 28-53 for more information.
- Up to 32 policy lists (including the default list) are supported per switch.
- Policy lists are not active on the switch until the **qos apply** command is issued.
- A rule may belong to a Universal Network Profile (UNP) list, the default list, and an egress policy list at the same time. In addition, a rule can also belong to multiple UNP or egress policy lists. Each time a rule is assigned to a policy list, however, an instance of that rule is created. Each instance is allocated system resources.
- If the rule is going to belong to a QoS policy list for a UNP, use the **no default-list** option when creating the rule. Doing so will give the rule precedence over default list rules when the policy list is applied to UNP device traffic.

- Only one policy list per UNP is allowed, but a single policy list can be associated with multiple profiles. See Chapter 30, "Configuring Access Guardian," for more information.
- A QoS policy list that is assigned to an Application Fingerprinting port must contain policy rules with the **appfp-group** condition.
- Only those rules that are assigned to an egress policy list are applied to egress traffic. When configuring egress policy lists, consider the following:
  - Egress policy lists are *not* supported on the OmniSwitch 6360, OmniSwitch 6465, and OmniSwitch 6560.
  - Only one egress policy list per switch is supported, to which IPv4 and IPv6 rules can be added.
  - Applying egress policy lists to SPB or VXLAN SAP ports is not supported.
  - Only the following policy conditions and actions are supported when creating rules for an egress policy list:

policy conditions	policy actions
Destination port Destination VLAN Source IPv4 address Source IPv6 address IPv6 (qualifier for traffic types)	Disposition (drop/accept)

  - On an OmniSwitch 9900, a destination port condition is supported only when the condition is part of a policy rule that was configured using the **no default-list** option (for example, **policy rule r1 condition c1 action a1 no default-list**) *and* the rule is assigned to an egress policy list.
- On the OmniSwitch 6360 and OmniSwitch 6465, policy rules containing the following conditions are not supported in a UNP policy list:
  - Source port group
  - Source IPv6 address
  - IPv6 flow label
  - Source MAC/MAC group
  - Destination MAC/ MAC group
  - Ethertype for IPV6
- On the OmniSwitch 6360, OmniSwitch 6560, and OmniSwitch 9900, only policy rules with the following conditions can be assigned to a UNP policy list:
  - Destination MAC
  - EtherType
  - Source VLAN
  - SIP
  - DIP / DIPv6
  - Layer 4 Protocol
  - Layer 4 source port
  - Layer 4 destination port
  - Source port bitmap

Using the Default Policy List

A default policy list always exists in the switch configuration. By default, a policy rule is added to this list at the time the rule is created. A rule remains a member a of the default list even when it is subsequently assigned to additional lists.

Each time a rule is assigned to a list, an instance of that rule is created and allocated system resources. As a result, rules that belong to multiple lists create multiple instances of the same rule. One way to conserve resources is to remove a rule from the default policy list.

To exclude a rule from the default policy list, use the **no default-list** option of the **policy rule** command when the rule is created. For example:

```
-> policy rule r1 condition c1 action a1 no default-list
```

The **no default-list** option can also remove an existing rule from the default list. For example, the **r2** rule already exists in the switch configuration but was not excluded from the default list at the time the rule was created. The following command removes the rule from the default list:

```
-> policy rule r2 condition c1 action a1 no default-list
```

To add an existing rule to the default list, use the **default-list** parameter option of the policy rule command. For example:

```
-> policy rule r2 condition c1 action a1 default-list
```

Rules associated with the default policy list are applied only to ingress traffic, unless the rule is also assigned to an egress policy list.

Verifying Policy Configuration

To view information about policy rules, conditions, and actions configured on the switch, use the following commands:

show policy condition	Displays information about all pending and applied policy conditions or a particular policy condition configured on the switch. Use the **applied** keyword to display information about applied conditions only.
show policy action	Displays information about all pending and applied policy actions or a particular policy action configured on the switch. Use the **applied** keyword to display information about applied actions only.
show policy rule	Displays information about all pending and applied policy rules or a particular policy rule. Use the **applied** keyword to display information about applied rules only.
show active policy rule	Displays applied policy rules that are active (enabled) on the switch.

Using Condition Groups in Policies

Condition groups are made up of multiple IPv4 addresses, MAC addresses, services, ports, or VLANs to which you want to apply the same action or policy rule. Instead of creating a separate condition for each address, etc., create a condition group and associate the group with a condition. Groups are especially useful when configuring filters, or Access Control Lists (ACLs); they reduce the number of conditions and rules that must be entered. For information about setting up ACLs, see Chapter 28, "Configuring QoS."

Commands used for configuring condition groups include the following:

policy network group
policy service group
policy mac group
policy port group

Access Control Lists (ACLs) typically use condition groups in policy conditions to reduce the number of rules required to filter particular types of traffic. For more information about ACLs, see "Using Access Control Lists" on page 28-64.

Sample Group Configuration

1 Create the group and group entries. In this example, a network group is created:

```
-> policy network group netgroup1 10.10.5.1 10.10.5.2
```

2 Attach the group to a policy condition. For more information about configuring conditions, see "Creating Policy Conditions" on page 28-45.

```
-> policy condition cond3 source network group netgroup1
```

Note. (*Optional*) Use the **show policy network group** command to display information about the network group. Each type of condition group has a corresponding show command. For example:

```
-> show policy network group
Group Name:                 From   Entries
Switch                      blt    4.0.1.166
10.0.1.166

+netgroup1                  cli    10.10.5.1/255.255.255.0
                                    10.10.5.2/255/255/255.0
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the output of this display. See "Verifying Condition Group Configuration" on page 28-60 for more information about using **show** commands to display information about condition groups.

---

3 Attach the condition to a policy rule. (For more information about configuring rules, see "Creating Policy Rules" on page 28-47.) In this example, action **act4** has already been configured. For example:

```
-> policy rule my_rule condition cond3 action act4
```

4 Apply the configuration. See "Applying the Configuration" on page 28-72 for more information about this command.

```
-> qos apply
```

Creating Network Groups

Use network policy groups for policies based on IPv4 source or destination addresses. The policy condition specifies whether the network group is a source network group, destination network group, or multicast network group.

- **Default switch group**—The switch contains a default network group called **switch** (includes IPv4 addresses configured for the switch) and a default network group called **switch6** (includes IPv6 addresses configured for the switch). Both network groups can also be used in policy conditions.
- **ACLs**—Typically network groups are used for Access Control Lists. For more information about ACLs, see "Using Access Control Lists" on page 28-64.

To create a network policy group, use the **policy network group** command. Specify the name of the group and the IP address(es) to be included in the group. Each IP address must be separated by a space. A mask can also be specified for an address. If a mask is not specified, the address is assumed to be a host address.

Note. Network group configuration is not active until the **qos apply** command is entered.

In this example, a policy network group called **netgroup2** is created with two IPv4 addresses. No mask is specified, so the IPv4 addresses are assumed to be host addresses.

```
-> policy network group netgroup2 10.10.5.1 10.10.5.2
```

In the next example, a policy network group called **netgroup3** is created with two IPv4 addresses. The first address also specifies a mask.

```
-> policy network group netgroup3 173.21.4.39 mask 255.255.255.0 10.10.5.3
```

In this example, the 173.201.4.39 address is subnetted, so that any address in the subnet is included in the network group. For the second address, 10.10.5.3, a mask is not specified; the address is assumed to be a host address.

The network group can then be associated with a condition through the **policy condition** command. The network group must be specified as a **source network group** or **destination network group**. In this example, **netgroup3** is configured for condition **c4** as source network group:

```
-> policy condition c4 source network group netgroup3
```

To remove addresses from a network group, use **no** and the relevant address(es). For example:

```
-> policy network group netgroup3 no 173.21.4.39
```

This command deletes the 173.21.4.39 address from **netgroup3** after the next **qos apply**.

To remove a network group from the configuration, use the **no** form of the **policy network group** command with the relevant network group name. The network group must not be associated with any policy condition or action. For example:

```
-> no policy network group netgroup3
```

If the network group is not currently associated with any condition or action, the network group **netgroup3** is deleted from the configuration after the next **qos apply**.

If a condition or an action is using **netgroup3**, the switch displays an error message similar to the following:

```
ERROR: netgroup3 is being used by condition 'c4'
```

In this case, remove the network group from the condition first, then enter the **no** form of the **policy network group** command. For example:

```
-> policy condition c4 no source network group
-> no policy network group netgroup3
```

The **policy condition** command removes the network group from the condition. (See "Creating Policy Conditions" on page 28-45 for more information about configuring policy conditions.) The network group is deleted at the next **qos apply**.

Creating Services

Policy services are made up of TCP or UDP ports or port ranges. They include source or destination ports, or both, but the ports must be the same type (TCP *or* UDP). Mixed port types cannot be included in the same service.

Policy services can be associated with policy service groups, which are then associated with policy conditions; or they can be directly associated with policy conditions.

To create a service, use the **policy service** command. With this command, there are two different methods for configuring a service. You can specify the protocol and the IP port; or you can use shortcut keywords. The following table lists the keyword combinations:

Procedure	Keywords	Notes
Basic procedure for either TCP or UDP service	**protocol** **source ip-port** **destination ip-port**	*The protocol must be specified with at least one source or destination port.*
Shortcut for TCP service	**source tcp-port** **destination tcp-port**	*Keywords can be used in combination.*
Shortcut for UDP service	**source udp-port** **destination udp-port**	*Keywords can be used in combination.*

An IP protocol (TCP or UDP), source IP port and/or destination IP port (or port range) must be associated with a service. IP port numbers are well-known port numbers defined by the IANA. For example, port numbers for FTP are 20 and 21; Telnet is 23.

In this example, a policy service called **telnet1** is created with the TCP protocol number (**6**) and the well-known Telnet destination port number (**23**).

```
-> policy service telnet1 protocol 6 destination ip-port 23
```

A shortcut for this command replaces the **protocol** and **destination ip-port** keywords with **destination tcp-port**:

```
-> policy service telnet1 destination tcp-port 23
```

In the next example, a policy service called **ftp2** is created with port numbers for FTP (20 and 21):

```
-> policy service ftp2 protocol 6 source ip-port 20-21 destination ip-port 20
```

A shortcut for this command replaces the **protocol**, **source ip-port**, and **destination ip-port** keywords with **source tcp-port** and **destination tcp-port**:

```
-> policy service ftp2 source tcp-port 20-21 destination tcp-port 20
```

Multiple services created through the **policy service** command can be associated with a policy service group; or, individual services can be configured for a policy condition. If you have multiple services to

associate with a condition, configure a service group and attach it to a condition. Service groups are described in "Creating Service Groups" on page 28-57.

Note. Service configuration is not active until the **qos apply** command is entered.

To remove a policy service, enter the **no** form of the command.

```
-> no policy service ftp2
```

The **ftp2** service is deleted from the configuration at the next **qos apply** if the service is not currently associated with a policy condition or a service group.

Creating Service Groups

Service groups are made up of policy services. First configure the policy service, then create the service group which includes the policy service(s).

Use the **policy service group** command. For example:

```
-> policy service group serv_group telnet1 ftp2
```

In this example, a policy service group called **serv_group** is created with two policy services (**telnet1** and **ftp2**). The policy services were created with the **policy service** command. (See "Creating Services" on page 28-56 for information about configuring policy services.)

Note. The policy service group can include only services with all source ports, all destination ports, or all source and destination ports. For example, the group cannot include a service that specifies a source port and another service that specifies a destination port.

The service group can then be associated with a condition through the **policy condition** command. For example:

```
-> policy condition c6 service group serv_group
```

This command configures a condition called **c6** with service group **serv_group**. All of the services specified in the service group are included in the condition. (For more information about configuring conditions, see "Creating Policy Conditions" on page 28-45.)

Note. Service group configuration must be specifically applied to the configuration with the **qos apply** command.

To delete a service from the service group, use **no** with the relevant service name. For example:

```
-> policy service group serv_group no telnet1
```

In this example, the service **telnet1** is removed from policy service group **serv_group**.

To delete a service group from the configuration, use the **no** form of the **policy service group** command. The service group must not be associated with any condition. For example:

```
-> no policy service group serv_group
```

Service group **serv_group** is deleted at the next **qos apply**. If **serv_group** is associated with a policy condition, an error message displays instead. For example:

```
ERROR: serv_group is being used by condition 'c6'
```

In this case, remove the service group from the condition first; then enter the **no policy service group** command. For example:

```
-> policy condition c6 no service group
-> no policy service group serv_group
```

The **policy condition** command removes the service group from the policy condition. (See "Creating Policy Conditions" on page 28-45 for more information about configuring policy conditions.) The service group is deleted at the next **qos apply**.

Creating MAC Groups

MAC groups are made up of multiple MAC addresses that you want to attach to a condition.

To create a MAC group, use the **policy mac group** command.

For example:

```
-> policy mac group macgrp2 08:00:20:00:00:00 mask ff:ff:ff:00:00:00
00:20:DA:05:f6:23
```

This command creates MAC group **macgrp2** with two MAC addresses. The first address includes a MAC address mask, so that any MAC address starting with 08:00:20 is included in **macgrp2**.

The MAC group can then be associated with a condition through the **policy condition** command. Note that the policy condition specifies whether the group must be used for *source* or *destination*. For example:

```
-> policy condition cond3 source mac group macgrp2
```

This command creates a condition called **cond3** that can be used in a policy rule to classify traffic by source MAC addresses. The MAC addresses are specified in the MAC group. For more information about configuring conditions, see "Creating Policy Conditions" on page 28-45.

Note. MAC group configuration is not active until the **qos apply** command is entered.

To delete addresses from a MAC group, use **no** and the relevant address(es):

```
-> policy mac group macgrp2 no 08:00:20:00:00:00
```

This command specifies that MAC address 08:00:20:00:00:00 is deleted from **macgrp2** at the next **qos apply**.

To delete a MAC group, use the **no** form of the **policy mac group** command with the relevant MAC group name. The group must not be associated with any policy condition. For example:

```
-> no policy mac group macgrp2
```

MAC group **macgrp2** is deleted at the next **qos apply.** If **macgrp2** is associated with a policy condition, an error message displays instead:

```
ERROR: macgrp2 is being used by condition 'cond3'
```

In this case, remove the MAC group from the condition first; then enter the **no policy mac group** command. For example:

```
-> policy condition cond3 no source mac group
-> no policy mac group macgrp2
```

The **policy condition** command removes the MAC group from the condition. See "Creating Policy Conditions" on page 28-45 for more information about configuring policy conditions. The MAC group is deleted at the next **qos apply**.

Creating Port Groups

Port groups are made up of slot and port number combinations. Note that there are many built-in port groups, one for each slot on the switch. Built-in port groups are subdivided by slice. The built in groups are named by slot (**Slot01**, **Slot02**, etc.). To view the built-in groups, use the **show policy port group** command.

To create a port group, use the **policy port group** command. For example:

```
-> policy port group techpubs 2/1/1 3/1/1 3/2/1 3/3/1
```

The port group can then be associated with a condition through the **policy condition** command. Note that the policy condition specifies whether the group must be used for *source* or *destination*. For example:

```
-> policy condition cond4 source port group techpubs
```

This command creates a condition called **cond4** that can be used in a policy rule to classify traffic by source port number. The port numbers are specified in the port group. For more information about configuring conditions, see "Creating Policy Conditions" on page 28-45.

Note. Port group configuration is not active until the **qos apply** command is entered.

To delete ports from a port group, use **no** and the relevant port number(s).

```
-> policy port group techpubs no 2/1/1
```

This command specifies that port 2/1 is deleted from the **techpubs** port group at the next **qos apply**.

To delete a port group, use the **no** form of the **policy port group** command with the relevant port group name. The port group must not be associated with any policy condition. For example:

```
-> no policy port group techpubs
```

The port group **techpubs** are deleted at the next **qos apply**. If **techpubs** is associated with a policy condition, an error message displays instead:

```
ERROR: techpubs is being used by condition 'cond4'
```

In this case, remove the port group from the condition first; then enter the **no policy port group** command. For example:

```
-> policy condition cond4 no source port group
-> no policy port group techpubs
```

The **policy condition** command removes the port group from the policy condition. (See "Creating Policy Conditions" on page 28-45 for more information about configuring policy conditions.) The port group is deleted at the next **qos apply**.

Verifying Condition Group Configuration

To display information about condition groups, use the following **show** commands:

show policy network group	Displays information about all pending and applied policy network groups or a particular network group. Use the **applied** keyword to display information about applied groups only.
show policy service	Displays information about all pending and applied policy services or a particular policy service configured on the switch. Use the **applied** keyword to display information about applied services only.
show policy service group	Displays information about all pending and applied policy service groups or a particular service group. Use the **applied** keyword to display information about applied groups only.
show policy mac group	Displays information about all pending and applied MAC groups or a particular policy MAC group configured on the switch. Use the **applied** keyword to display information about applied groups only.
show policy port group	Displays information about all pending and applied policy port groups or a particular port group. Use the **applied** keyword to display information about applied groups only.

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the syntax and output for these commands.

Using Map Groups

Map groups are used to map 802.1p, ToS, or DSCP values to different values. The following mapping scenarios are supported:

- 802.1p to 802.1p, based on Layer 2, Layer 3, and Layer 4 parameters and source/destination slot/port. In addition, 802.1p classification can trigger this action.
- ToS or DSCP to 802.1p, based on Layer 3 and Layer 4 parameters and source/destination slot/port. In addition ToS or DSCP classification can trigger this action.

Note. Map groups are associated with a policy *action*.

Commands used for creating map groups include the following:

policy map group
policy action map

Sample Map Group Configuration

1 Create the map group with mapping values. For detailed information about map groups and how to set them up, see "How Map Groups Work" on page 28-62 and "Creating Map Groups" on page 28-62.

```
-> policy map group tosGroup 1-2:5 4:5 5-6:7
```

2 Attach the map group to a policy action. See "Creating Policy Actions" on page 28-46 for more information about creating policy actions.

```
-> policy action tosMap map tos to 802.1p using tosGroup
```

Note. (*Optional*) Use the **show policy map group** command to verify the map group.

```
-> show policy map group
Group Name                    From  Entries
+tosGroup                      cli  1-2:5
                                    4:5
                                    5-6:7
```

For more information about this command, see "Verifying Map Group Configuration" on page 28-63 and the *OmniSwitch AOS Release 8 CLI Reference Guide*.

3 Attach the action to a policy rule. In this example, the condition **Traffic** is already configured. For more information about configuring rules, see "Creating Policy Rules" on page 28-47.

```
-> policy rule r3 condition Traffic action tosMap
```

4 Apply the configuration. For more information about this command, see "Applying the Configuration" on page 28-72.

```
-> qos apply
```

How Map Groups Work

When mapping from 802.1p to 802.1p, the action results in remapping the specified values. Any values that are not specified in the map group are preserved. In this example, a map group is created for 802.1p bits.

```
-> policy map group Group2 1-2:5 4:5 5-6:7
-> policy action Map1 map 802.1p to 802.1p using Group2
```

The *to* and *from* values are separated by a colon (**:**). If traffic with 802.1p bits comes into the switch and matches a policy that specifies the **Map1** action, the bits are remapped according to **Group2**. If the incoming 802.1p value is 1 or 2, the value is mapped to 5. If the incoming 802.1p value is 3, the outgoing value is 3 (the map group does not specify any mapping for a value of 3). If the incoming 802.1p value is 4, the value is mapped to 5. If the incoming 802.1p value is 5 or 6, the value is mapped to 7.

When mapping to a different type of value, however (ToS/DSCP to 802.1p), any values in the incoming flow that matches the rule but that are not included in the map group is zeroed out. For example, the following action specifies the same map group but instead specifies mapping 802.1p to ToS:

```
-> policy action Map2 map tos to 802.1p using Group2
```

In this case, if ToS traffic comes into the switch and matches a policy that specifies the **Map2** action, the ToS value is mapped according to **Group2** if the value is specified in **Group2**. If the incoming ToS value is 2, the value is mapped to 5; however, if the incoming value is 3, the switch maps the value to zero because there is no mapping in **Group2** for a value of 3.

Note. Ports on which the flow is mapped must be a trusted port; otherwise the flow is dropped.

Creating Map Groups

To create a map group, use the **policy action map** command. For example, to create a map group called **tosGroup**, enter:

```
-> policy map group tosGroup 1-2:5 4:5 5-6:7
```

The to and from values are separated by a colon (**:**). For example, a value of 2 is mapped to 5.

Note. Map group configuration is not active until the **qos apply** command is entered.

The remapping group can then be associated with a rule through the **policy action** command. In this example, a policy condition called **Traffic** has already been configured.

```
-> policy action tosMap map tos to 802.1p using tosGroup
-> policy rule r3 condition Traffic action tosMap
```

To delete mapping values from a group, use **no** and the relevant values:

```
-> policy map group tosGroup no 1-2:4
```

The specified values are deleted from the map group at the next **qos apply**.

To delete a map group, use the **no** form of the **policy map group** command. The map group must not be associated with a policy action. For example:

```
-> no policy map group tosGroup
```

If **tosGroup** is currently associated with an action, an error message similar to the following displays:

```
ERROR: tosGroup is being used by action 'tosMap'
```

In this case, remove the map group from the action, then enter the **no policy map group** command:

```
-> policy action tosMap no map group
-> no policy map group tosGroup
```

The map group is deleted at the next **qos apply**.

Note. For Layer 2 flows, you cannot have more than one action that maps DSCP.

Verifying Map Group Configuration

To display information about all map groups, including all pending and applied map groups, use the **show policy map group** command. To display only information about applied map groups, use the **applied** keyword with the command. For more information about the output of this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Using Access Control Lists

Access Control Lists (ACLs) are QoS policies used to control whether or not packet flows are allowed or denied at the switch or router interface. ACLs are sometimes referred to as filtering lists.

ACLs are distinguished by the kind of traffic they filter. In a QoS policy rule, the type of traffic is specified in the policy condition. The policy action determines whether the traffic is allowed or denied. For detailed descriptions about configuring policy rules, see "QoS Policy Overview" on page 28-29 and "Creating Policies" on page 28-43.

In general, the types of ACLs include:

- *Layer 2 ACLs*—for filtering traffic at the MAC layer. Usually uses MAC addresses or MAC groups for filtering.
- *Layer 3/4 ACLs*—for filtering traffic at the network layer. Typically uses IP addresses or IP ports for filtering; note that IPX filtering is not supported.
- *Multicast ACLs*—for filtering IGMP traffic.
- *Security ACLs*—for improving network security. These ACLs utilize specific security features, such as **UserPorts** groups to prevent source IP address spoofing, ICMP drop rules, and TCP connection rules.

Layer 2 ACLs

Layer 2 filtering filters traffic at the MAC layer. Layer 2 filtering can be done for both bridged and routed packets. As MAC addresses are learned on the switch, QoS classifies the traffic based on:

- MAC address or MAC group
- Source VLAN
- Physical slot/port or port group

The switch classifies the MAC address as both source ***and*** destination.

Layer 2 ACL: Example 1

This example configures an ACL policy rule that is used to filter traffic from a specific source MAC address learned on a specific VLAN.

```
-> policy condition Address1 source mac 08:00:20:11:22:33 source vlan 5
-> policy action BlockTraffic disposition deny
-> policy rule FilterA condition Address1 action BlockTraffic
```

In this scenario, traffic with a source MAC address of 08:00:20:11:22:33 coming in on VLAN 5 would match condition **Address1**, which is a condition for a policy rule called **FilterA**. **FilterA** is then applied to the flow. Since **FilterA** has an action (**BlockTraffic**) that is set to deny traffic, the flow would be denied on the switch.

Note that although this example contains only Layer 2 conditions, it is possible to combine Layer 2 and Layer 3 conditions in the same policy.

Layer 2 ACL: Example 2

Maintaining the 802.1p Priority for IP Packets

When a tagged IP packet ingresses on a trusted port and the default classification priority for that port is set to DSCP (using the default DSCP value of 0), the DSCP value of the packet is mapped to the 802.1p value of the same packet. To avoid overwriting the 802.1p value in this scenario, configure an ACL as follows:

1. Create a port group to include all of the ports that QoS must trust.
2. Define policy conditions for the port group; one condition for each L2 priority (802.1p) value.
3. Define policy actions that stamp the IP traffic with the L2 priority value.
4. Define policy rules using the conditions and actions created in Steps 2 and 3.
5. Do not globally trust all switch ports.

For example:

```
-> policy port group VoIP 1/1/4-6 1/1/8 1/2/3-5
-> policy condition p0 destination port group VoIP
-> policy condition p1 destination port group VoIP
-> policy condition p2 destination port group VoIP
-> policy condition p3 destination port group VoIP
-> policy condition p4 destination port group VoIP
-> policy condition p5 destination port group VoIP
-> policy condition p6 destination port group VoIP
-> policy condition p7 destination port group VoIP
-> policy action p0 802.1p 0
-> policy action p1 802.1p 1
-> policy action p2 802.1p 2
-> policy action p3 802.1p 3
-> policy action p4 802.1p 4
-> policy action p5 802.1p 5
-> policy action p6 802.1p 6
-> policy action p7 802.1p 7
-> policy rule p0 condition p0 action p0
-> policy rule p1 condition p1 action p1
-> policy rule p2 condition p2 action p2
-> policy rule p3 condition p3 action p3
-> policy rule p4 condition p4 action p4
-> policy rule p5 condition p5 action p5
-> policy rule p6 condition p6 action p6
-> policy rule p7 condition p7 action p7
-> qos apply
```

Note. For pure Layer 2 packets, trusted ports retain the 802.1p value of the packet and queue the packets according to that priority value.

Layer 3 ACLs

The QoS software in the switch filters routed and bridged traffic at Layer 3.

For Layer 3 filtering, the QoS software in the switch classifies traffic based on:

- Source IP address or source network group
- Destination IP address or destination network group
- IP protocol
- ICMP code
- ICMP type
- Source TCP/UDP port
- Destination TCP/UDP port or service or service group

Layer 3 ACL: Example 1

This example configures an ACL policy rule that is used to filter a specific flow of IP traffic.

```
-> policy condition addr2 source ip 192.68.82.0 source ip-port 23 ip-protocol 6
-> policy action Block disposition deny
-> policy rule FilterL31 condition addr2 action Block
```

Traffic with a source IP address of 192.68.82.0, a source IP port of 23, using protocol 6, matches condition **addr2**, which is part of **FilterL31**. The action for the filter (**Block**) is set to deny traffic. The flow is dropped on the switch.

Note that although this example contains only Layer 3 conditions, it is possible to combine Layer 2 and Layer 3 conditions in the same policy.

Layer 3 ACL: Example 2

This example uses condition groups to combine multiple IP addresses in a single condition.

```
-> policy network group GroupA 192.60.22.1 192.60.22.2 192.60.22.0
-> policy condition cond7 destination network group GroupA
-> policy action Ok disposition accept
-> policy rule FilterL32 condition cond7 action Ok
```

In this example, a network group, **GroupA**, is configured with three IP addresses. Condition **cond7** includes **GroupA** as a destination group. Flows coming into the switch destined for any of the specified IP addresses in the group matches rule **FilterL32**. **FilterL32** is configured with an action (**Ok**) to allow the traffic on the switch.

Note that although this example contains only a Layer 3 condition, it is possible to combine Layer 2 and Layer 3 conditions in the same policy.

IPv6 ACLs

An ACL is considered an IPv6 ACL if the **ipv6** keyword and/or any of the following specific policy condition keywords are used in the ACL to classify/filter IPv6 traffic:

IPv6 ACL Keywords	
source mac	**802.1p**
source mac group	**flow-label**
destination mac	**icmpcode**
destination mac group	**icmptype**
source port	**service**
source port group	**service group**
source tcp-port	**tcpflags**
destination tcp-port	**tos**
source udp-port	**nh (next header)**
destination udp-port	
source ipv6	
source network group	
destination ipv6	
destination network group	

Note that IPv6 ACLs are effected only on IPv6 traffic. All other ACLs/policies with IP conditions that do not use the **ipv6** keyword are effected only on IPv4 traffic. For example:

```
-> policy condition c1 tos 7
-> policy condition c2 tos 7 ipv6
```

In the above example, c1 is an IPv4 condition and c2 is an IPv6 condition. ACLs that use c1 are considered IPv4 policies; ACLs that use c2 are considered IPv6 policies. In addition, consider the following examples:

```
-> policy condition c3 source port 1/10
-> policy condition c4 source port 1/10 ipv6
```

Condition c3 applies to all traffic ingressing on port 1/10. However, condition c4 applies only to IPv6 traffic ingressing on port 1/10.

Consider the following guidelines when configuring IPv6 ACLs:

- Trusted/untrusted behavior is the same for IPv6 traffic as it is for IPv4 traffic.
- On the OmniSwitch 6560 and OmniSwitch 9900, the following source IPv6 address policy conditions are supported only for *egress* IPv6 ACLs:
  - **source ipv6**
  - **source network group** with an IPv6 address (includes user-configured and the built-in “Switch6” group)
- On the OmniSwitch 6560 and OmniSwitch 9900, the following destination IPv6 address policy conditions are supported only for *ingress* IPv6 ACLs:
  - **destination ipv6**
  - **destination network group** with an IPv6 address
- IPv6 policies do not support the use of map groups.
- IPv6 multicast policies are not supported.
- Anti-spoofing and other UserPorts profiles/filters do not support IPv6.

- The default (built-in) network group, “Switch6”, only applies to IPv6 addresses configured for the switch.
- The default (built-in) network group, “Switch”, only applies to IPv4 addresses configured for the switch.

Multicast Filtering ACLs

Multicast filtering can be set up to filter clients requesting group membership through the Internet Group Management Protocol (IGMP). IGMP is used to track multicast group membership. The IP Multicast Switching (IPMS) function in the switch optimizes the delivery of IP multicast traffic by sending packets only to those stations that request it. Potential multicast group members can be filtered out so that IPMS does not send multicast packets to those stations.

For more information about IPMS, see Chapter 27, “Configuring IP Multicast Switching.”

The global disposition for multicast traffic is set to accept. For multicast filtering, the switch classifies traffic based on the multicast IP address or multicast network group and any destination parameters. Note that the destination parameters are used for the client from which the switch receives the IGMP request.

The **multicast ip** or **multicast network group** keyword is required in the condition configured for a multicast ACL.

The following keywords can be used in the condition to indicate the client parameters:

Multicast ACL Keywords
destination ip
destination vlan
destination port
destination port group
destination mac
destination mac group

If a destination group is specified, the corresponding single value keyword cannot be combined in the same condition. For example, if a destination port is specified, a destination port group cannot be specified in the same condition.

To filter multicast clients, specify the multicast IP address, which is the address of the multicast group or stream, and specify the client IP address, VLAN, MAC address, or slot/port. For example:

```
-> policy condition Mclient1 multicast ip 225.0.1.2 destination vlan 5
-> policy action ok disposition accept
-> policy rule Mrule condition Mclient1 action ok
```

In this example, any traffic coming in on VLAN 5 requesting membership to the 225.0.1.2 multicast group is allowed to pass through.

Using ACL Security Features

The following additional ACL features are available for improving network security and preventing malicious activity on the network:

- **UserPorts**—A port group that identifies its members as user ports to prevent source address spoofing of IP and ARP traffic (per RFC 2267). When a port is configured as a member of this group, packets received on the port are dropped if they contain a source IP address that does not match the IP subnet for the port. It is also possible to configure a UserPorts profile to specify other types of traffic to

monitor on user ports. See "Configuring a UserPorts Group" on page 28-69.

- **ICMP drop rules**—Allows condition combinations in policies that prevent user pings, thus reducing DoS exposure from pings. Two condition parameters are also available to provide more granular filtering of ICMP packets: **icmptype** and **icmpcode**. See "Configuring ICMP Drop Rules" on page 28-70.
- **TCP connection rules**—Allows the determination of an *established* TCP connection by examining TCP flags found in the TCP header of the packet. Two condition parameters are available for defining a TCP connection ACL: **established** and **tcpflags**. See "Configuring TCP Connection Rules" on page 28-70.
- **Early ARP discard**—ARP packets destined for other hosts are discarded to reduce processing overhead and exposure to ARP DoS attacks. No configuration is required to use this feature, it is always available and active on the switch. Note that ARPs intended for use by a local subnet, AVLAN, VRRP, and Local Proxy ARP are *not* discarded.
- **ARP ACLs**—It is also possible to create an ACL that examines the source IP address in the header of ARP packets. This is done by specifying the ARP ethertype (0x0806) and source IP address.

Configuring a UserPorts Group

To prevent IP address spoofing and/or other types of traffic on specific ports, create a port group called **UserPorts** and add the ports to that group. For example, the following **policy port group** command adds ports 1/1-24, 2/1-24, 3/1, and 4/1 to the **UserPorts** group:

```
-> policy port group UserPorts 1/1-24 2/1-24 3/1 4/1
-> qos apply
```

Note that the UserPorts group applies to both bridged and routed traffic, and it is *not* necessary to include the UserPorts group in a condition and/or rule for the group to take effect. Once ports are designated as members of this group, IP spoofed traffic is blocked while normal traffic is still allowed on the port.

Configuring UserPort Traffic Types and Port Behavior

In addition to spoofed traffic, it is also possible to configure QoS to look for BPDU, RIP, OSPF, BGP, VRRP, and/or DHCP server packets on user ports. When the specified type of traffic is encountered, the user port can either filter the traffic or administratively shutdown to block all traffic.

Consider the following when configuring the type of traffic and port behavior that is applied to ports assigned to the UserPorts group:

- The **qos user-port** command is used to configure a UserPorts profile that specifies the types of traffic to look for and select how the ports will deal with such traffic.
- A slot and port number is not required with the **qos user-port** command. This is because the command applies to all ports that are members of the UserPorts group.
- Ingress traffic is filtered on ports that are members of the UserPorts group. However, the switch will still process the filtered packets to determine if an egress update is sent on the same port. For example, if RIP traffic is filtered, the switch will still send RIP peer updates on that port.
- An SNMP trap is sent whenever a user port shutdown occurs. To enable a port disabled by a user port shutdown operation, use the **interfaces** command to administratively enable the port or disconnect and reconnect the port cable.
- Any changes to the UserPorts profile (for example, adding or removing a traffic type) are not made until the **qos apply** command is performed.

By default, spoofed traffic is filtered on user ports. To change the types of traffic filtered, use the **qos user-port** command with the **filter** option. For example, the following command specifies that user ports must filter BPDU packets:

```
-> qos user-port filter bpdu
```

To specify multiple types of traffic on the same command line, enter each type separated by a space. For example:

```
-> qos user-port filter ospf bgp rip
```

Each time the **qos user-port** command is used, any traffic types that were previously configured are removed. To retain the previous configuration, specify all of the desired traffic types each time the **qos user-port** command is performed. For example, the following command filters spoofed and BPDU traffic:

```
-> qos user-port filter spoof bpdu
```

To add filtering for RIP traffic and retain the filtering configuration for spoofed and BPDU traffic, specify all three types of traffic. For example:

```
-> qos user-port filter spoof bpdu rip
```

In the above command example, if **spoof** and **bpdu** were *not* specified, then the switch would only filter RIP traffic.

The following **qos user-port** command example uses the **shutdown** option to administratively disable the user port if the specified type of traffic is received on that port:

```
-> qos user-port shutdown bpdu
```

To disable the filter or shutdown function, use the **no** form of the **qos user-port** command. For example, the following command disables the filtering operation for all user ports:

```
-> qos no user-port filter
```

Use the **show qos config** command to display the **qos user-port** command settings.

Configuring ICMP Drop Rules

Combining a Layer 2 condition for source VLAN with a Layer 3 condition for IP protocol is supported. In addition, two new condition parameters are available to provide more granular filtering of ICMP packets: **icmptype** and **icmpcode**. Use these two conditions together in a policy to block ICMP echo request and reply packets without impacting switch performance.

The following example defines an ACL policy that prevents users from pinging by dropping echo request ICMP packets at the source port:

```
-> policy condition pingEchoRequest source vlan 10 icmptype 8
-> policy action drop disposition drop
-> policy rule noping10 condition pingEchoRequest action drop
-> qos apply
```

Note that the above policy only blocks ICMP echo traffic, all other ICMP traffic is still allowed.

Configuring TCP Connection Rules

Two condition parameters are available for defining a TCP connection ACL policy: **established** and **tcpflags**. An ACL can be defined using the **established** parameter to identify packets that are part of an established TCP connection and allow forwarding of the packets to continue. When this parameter is

invoked, TCP header information is examined to determine if the **ack** or **rst** flag bit is set. If this condition is true, then the connection is considered established.

The following is an example ACL policy using the **established** condition parameter:

```
policy condition c destination ip 192.168.10.0 mask 255.255.255.0 established
policy condition c1 destination ip 192.168.10.0 mask 255.255.255.0
policy action drop disposition drop
policy action allow

policy rule r condition c action allow
policy rule r1 condition c1 action drop
qos apply
```

This example ACL policy prevents any TCP connection from being initiated to the 192.168.10.0 network and all other IP traffic to the 192.168.10.0 network. Only TCP connections initiated from the 192.168.10.0 network are allowed.

Note that the above example ACL would prevent FTP sessions. See the **policy condition established** command page in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information.

An ACL can also be defined using the **tcpflags** parameter to examine and qualify specific TCP flags individually or in combination with other flags. This parameter can be used to prevent specific DOS attacks, such as the *christmas tree*.

The following example use the **tcpflags** condition parameter to determine if the F (fin) and S (syn) TCP flag bits are set to one and the A (ack) bit is set to zero:

```
-> policy condition c1 tcpflags all f s mask f s a
```

In this example, a match must occur on all the flags or the packet is not allowed. If the optional command keyword **any** was used, then a match need only occur on any one of the flags. For example, the following condition specifies that either the A (ack) bit or the R (rst) bit must equal one:

```
-> policy condition c1 tcpflags any a r mask a r
```

Note that if a flag is specified on the command line after the **any** or **all** keyword, then the match value is one. If the flag only appears as part of the **mask**, then the match value is zero. See the **policy condition tcpflags** command page in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information.

Although there are two condition parameters available to define a TCP connection policy, only use one or the other in the same policy condition. For example, the following command attempts to configure a policy condition using both the **tcpflags** and **established** parameters:

```
-> policy condition tcpflag-f tcpflags any f mask f established ipv6
ERROR: tcpflag-f: Can't use 'established' with 'tcpflags'
```

Applying the Configuration

Configuration for policy rules and many global QoS parameters must specifically be applied to the configuration with the **qos apply** command. Any parameters configured without this command are maintained for the current session but are not yet activated. For example, if you configure a new policy rule through the **policy rule** command, the switch cannot use it to classify traffic and enforce the policy action until the **qos apply** command is entered. For example:

```
-> policy rule my_rule condition c4 action a5
-> qos apply
```

The **qos apply** command must be included in an ASCII text configuration file when QoS commands are included. The command must be included after the last QoS command.

When the configuration is not yet applied, it is referred to as the *pending configuration*.

Global Commands. Many global QoS commands are active immediately on the switch *without* **qos apply**. *The settings configured by these commands become active immediately.* Other global commands must specifically be applied. The commands are listed in the following table:

Global Commands That Take Effect Immediately	
qos	**qos trust ports**
qos forward log	**qos stats interval**
qos log console	**qos revert**
qos log lines	**qos flush**
qos log level	**qos reset**
debug qos	

Port and Policy Commands. All port parameters and policy parameters must be applied with the **qos apply** command.

Port and Policy Commands	
qos port	**policy service**
policy condition	**policy service group**
policy action	**policy mac group**
policy rule	**policy port group**
policy network group	**policy map group**

The pending configuration is useful for reviewing policy rules before actually applying them to the switch.

Applied policy rules can also be administratively disabled (inactive). If a rule is administratively disabled, the rule exists in the applied configuration but does not be used to classify flows. For more information about disabling/re-enabling a policy rule, see "Creating Policy Rules" on page 28-47.

Deleting the Pending Configuration

Policy settings that have been configured but not applied through the **qos apply** command can be returned to the last applied settings through the **qos revert** command. For example:

```
-> qos revert
```

This command ignores any pending policies (any additions, modifications, or deletions to the policy configuration since the last **qos apply)** and writes the last applied policies to the pending configuration. At this point, the pending policies are the same as the last applied policies.

In this example, there are two new pending policies and three applied policies:

Pending Policies	Applied Policies
rule5 rule6	rule1 rule2 rule3

If you enter **qos revert**, the configuration then looks like:

Pending Policies	Applied Policies
rule1 rule2 rule3	rule1 rule2 rule3

Flushing the Configuration

In some cases, when you need to remove all of your rules and start over again, erase the pending policies completely from the configuration, use the **qos flush** command. For example:

```
-> qos flush
```

If you then enter **qos apply**, all policy information is deleted.

In this example, there are two new pending policies and three applied policies:

Pending Policies	Applied Policies
rule5 rule6	rule1 rule2 rule3

If you enter **qos flush**, the configuration then looks like:

Pending Policies	Applied Policies
	rule1 rule2 rule3

In this scenario, you can do one of two things. To write the applied policies back to the pending configuration, use **qos revert**. Or, to delete all policy rule configuration, enter **qos apply**. If **qos apply** is entered, the empty set of pending policies are written to the applied policies and all policy rule configuration is deleted.

Interaction With LDAP Policies

The **qos apply**, **qos revert**, and **qos flush** commands do not affect policies created through the PolicyView application. Separate commands are used for loading and flushing LDAP policies on the switch. See Chapter 33, “Managing Authentication Servers,” for information about managing LDAP policies.

Verifying the Applied Policy Configuration

The policy **show** commands have an optional keyword (**applied**) to display only applied policy objects. These commands include:

show policy condition	Displays information about all pending and applied policy conditions or a particular policy condition configured on the switch. Use the **applied** keyword to display information about applied conditions only.
show policy action	Displays information about all pending and applied policy actions or a particular policy action configured on the switch. Use the **applied** keyword to display information about applied actions only.
show policy rule	Displays information about all pending and applied policy rules or a particular policy rule. Use the **applied** keyword to display information about applied rules only.
show policy network group	Displays information about all pending and applied policy network groups or a particular network group. Use the **applied** keyword to display information about applied groups only.
show policy service	Displays information about all pending and applied policy services or a particular policy service configured on the switch. Use the **applied** keyword to display information about applied services only.
show policy service group	Displays information about all pending and applied policy service groups or a particular service group. Use the **applied** keyword to display information about applied groups only.
show policy mac group	Displays information about all pending and applied MAC groups or a particular policy MAC group configured on the switch. Use the **applied** keyword to display information about applied groups only.
show policy port group	Displays information about all pending and applied policy port groups or a particular port group. Use the **applied** keyword to display information about applied groups only.
show policy map group	Displays information about all pending and applied policy map groups or a particular map group. Use the **applied** keyword to display information about applied groups only.

For more information about these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Policy Applications

Policies are used to classify incoming flows and treat the relevant outgoing flows. There are many ways to classify the traffic and many ways to apply QoS parameters to the traffic.

Classifying traffic can be as simple as identifying a Layer 2 or Layer 3 address of an incoming flow. Treating the traffic might involve prioritizing the traffic or rewriting an IP address. How the traffic is treated (the *action* in the policy rule) typically defines the type of policy:

Type of Policy	Description	Action Parameters Used
Basic QoS policies	Prioritizes and polices particular flows.	**maximum bandwidth** **maximum depth** **priority** **cir cbs pir pbs**
Redirection policies	Redirects flows to a specific port or link aggregate ID.	**redirect port** **redirect linkagg**
Policy Based Mirroring	Mirrors ingress and egress packets to a specific port.	**ingress mirror** **egress mirror** **ingress egress mirror**
ICMP policies	Filters, prioritizes, and/or rate limits ICMP traffic	**disposition** **priority** **maximum bandwidth**
802.1p, ToS, and DSCP tagging or mapping policies	Sets or resets the egress 802.1p, ToS, or DSCP values	**802.1p** **tos** **dscp** **map group**
Policy Based Routing (PBR)	Redirects routed traffic.	**permanent gateway-ip** **permanent gateway-ipv6**
Access Control Lists (ACLs)	Groups of policies rules used for filtering traffic (allow/deny)	**disposition**

This section describes how to configure basic QoS policies and 802.1p/ToS/DSCP marking and mapping policies. Policies used for Layer 2 and Layer 3/4 filters, are commonly referred to as Access Control Lists (ACLs). Filtering is discussed in Chapter 28, “Configuring QoS.”

Policies can also be used for prioritizing traffic in dynamic link aggregation groups. For more information about dynamic link aggregates, see Chapter 10, “Configuring Dynamic Link Aggregation.”

Basic QoS Policies

Traffic prioritization and bandwidth policing can be the most common types of QoS policies. For these policies, any condition can be created; the policy action indicates how the traffic must be prioritized or how the bandwidth must be shaped.

Note. If multiple addresses, services, or ports must be given the same priority, use a policy condition group to specify the group and associate the group with the condition. See "Using Condition Groups in Policies" on page 28-54 for more information about groups.

Note that some condition parameters can be used in combination only under particular circumstances; also, there are restrictions on condition/action parameter combinations. See "Policy Conditions" on page 28-31 and "Policy Actions" on page 28-32.

Basic Commands

The following **policy action** commands are used for traffic prioritization or policing (rate limiting):

policy action priority
policy action maximum bandwidth
policy action maximum depth

To set up traffic prioritization and/or bandwidth policing, follow the steps in the next section. For more information about command syntax and options, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Note that QoS ports can also be configured for bandwidth shaping through the **qos port maximum ingress-bandwidth** and **qos port maximum egress-bandwidth** commands.

Traffic Prioritization Example

In this example, IP traffic is routed from the 10.10.4.0 network through the OmniSwitch.

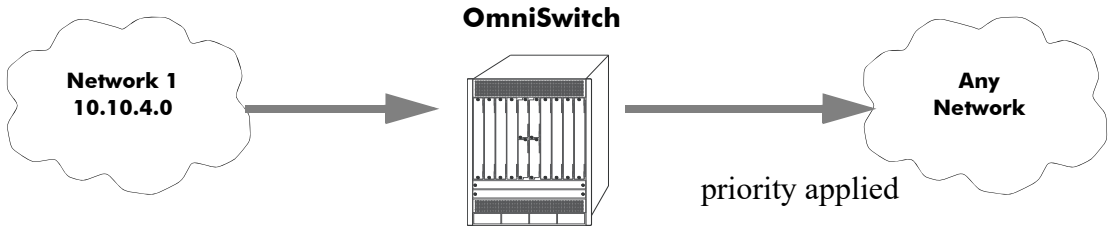


Figure 28-4 : Traffic Prioritization Example

To create a policy rule to prioritize the traffic from Network 1, first create a condition for the traffic that you want to prioritize. In this example, the condition is called **ip_traffic**. Then create an action to prioritize the traffic as highest priority. In this example, the action is called **high**. Combine the condition and the action into a policy rule called **rule1**.

```
-> policy condition ip_traffic source ip 10.10.4.0 mask 255.255.255.0
-> policy action high priority 7
-> policy rule rule1 condition ip_traffic action high
```

The rule is not active on the switch until the **qos apply** command is entered on the command line. When the rule is activated, any flows coming into the switch from 10.10.4.0 is given the highest priority.

Bandwidth Policing Example

In this example, a maximum bandwidth rate is effected on flows from a specific source IP address.

First, create a condition for the traffic. In this example, the condition is called **ip_traffic2**. A policy action (**flowShape**) is then created to enforce a maximum bandwidth requirement for the flow.

```
-> policy condition ip_traffic2 source ip 10.10.5.3
-> policy action flowShape maximum bandwidth 10m
-> policy action burst maximum depth 1m
-> policy rule rule2 condition traffic2 action flowShape action burst
```

Note that the bandwidth can be specified in abbreviated units, in this case, **1k**. The rule is not active on the switch until the **qos apply** command is entered.

Redirection Policies

A redirection policy sends traffic that matches the policy to a specific port or link aggregate instead of the originally intended destination. This type of policy can use any condition; the policy action determines which port or link aggregate to which the traffic is sent.

The following **policy action** commands are used for port and link aggregate redirection:

policy action redirect port
policy action redirect linkagg

Note the following regarding the use and configuration of redirection policies:

- Redirection policies apply to both bridged and routed traffic.
- When redirecting routed traffic from VLAN A to VLAN B, the redirect port or link aggregate ID must belong to VLAN B (tagged or default VLAN).
- Routed packets (from VLAN A to VLAN B) are not modified after they are redirected; the source and MAC address remain the same. In addition, if the redirect port or link aggregate ID is tagged, the redirected packets have a tag from the ingress VLAN A.
- If a route exists for the redirected flow, then redirected packets are the final post-routing packets.
- If a route does not exist for the redirected flow, the flow is not redirected to the specified port or link aggregate ID and is “blackholed”. As soon as a route is available, the flow is then redirected as specified in the policy.
- In most cases, a redirected flow does *not* trigger an update to the routing and ARP tables. When the ARP table is cleared or timed out, port/link aggregate redirection cease until the ARP table is refreshed. If necessary, create a static route for the flow or assign the redirect port or link aggregate ID to the ingress VLAN (VLAN A) to send packets to the redirect port until a route is available.
- When redirecting bridged traffic on VLAN A, the redirect port or link aggregate ID must belong to VLAN A (tagged or default VLAN).

In the following example, flows destined for UDP port 80 is redirected to switch port 3/2:

```
-> policy condition L4PORTCOND destination udp-port 80
-> policy action REDIRECTPORT redirect port 3/2
-> policy rule L4PORTRULE condition L4PORTCOND action REDIRECTPORT
```

In the following example, flows destined for IP address 40.2.70.200 are redirected to link aggregate 10:

```
-> policy condition L4LACOND destination IP 40.2.70.200
-> policy action REDIRECTLA redirect linkagg 10
-> policy rule L4LARULE condition L4LACOND action REDIRECTLA
```

Note that in both examples above, the rules are not active on the switch until the **qos apply** command is entered on the command line.

Policy Based Mirroring

A mirroring policy sends a copy of ingress, egress, or both ingress and egress packets that match the policy condition to a specific port. This type of policy can use any condition; the mirror policy action determines the type of traffic to mirror and the port on which the mirrored traffic is received.

The **policy action mirror** command is used to configure mirror-to-port (MTP) action for the policy. For example, the following policy mirrors ingress packets to port 1/10:

```
-> policy condition c1 source ip 192.168.20.1
-> policy action a1 ingress mirror 1/1/10
-> policy rule r1 condition c1 action a1
-> qos apply
```

When the above rule is activated, any flows coming into the switch from source IP address 192.168.20.1 are mirrored to port 1/10. It is also possible to combine the MTP action with other actions. For example:

```
-> policy condition c1 source ip 192.168.20.1
-> policy action a1 ingress mirror 1/1/10 disposition drop
-> policy rule r1 condition c1 action a1
-> qos apply
```

This policy rule example combines the MTP action with the drop action. As a result, this rule drops ingress traffic with a source IP of 192.168.20.1, but the mirrored traffic from this source is not dropped and is forwarded to port 1/10.

To send the mirror traffic to multiple destination ports, use the **policy action mirror session** command. The port mirroring session can contain multiple mirroring destinations or link aggregates. For example, the following policy mirrors the packets with source IP as 1.1.1.1 to all the ports that are a part of port-mirroring session 1.

```
-> policy condition c1 source ip 1.1.1.1
-> policy action a1 mirror session 1
-> policy rule r1 condition c1 action a1
-> qos apply
```

Note the following regarding the use and configuration of mirroring policies:

- Only one policy-based MTP session is supported at any given time either port-based policy mirroring or session-based policy mirroring. As a result, all mirroring policies must specify the same destination port or the same port mirroring session ID.
- In addition to one policy-based MTP session, the switch can support one port-based mirroring session, one remote port mirroring session, and one port monitoring session all running at the same time.
- Policy based mirroring and the port-based mirroring feature can run simultaneously on the same port.

- Rule precedence is applied to all mirroring policies that are configured for the same switch ASIC. If traffic matches a mirror rule on one ASIC with a lower precedence than a non-mirroring rule on a different ASIC, the traffic is mirrored in addition to the actions specified by the higher precedence rule.

ICMP Policy Example

Policies can be configured for ICMP on a global basis on the switch. ICMP policies can be used for security (for example, to drop traffic from the ICMP blaster virus).

In the following example, a condition called **icmpCondition** is created with no other condition parameters:

```
-> policy condition icmpCondition ip-protocol 1
-> policy action icmpAction disposition deny
-> policy rule icmpRule condition icmpCondition action icmpAction
```

This policy (**icmpRule**) drops all ICMP traffic. To limit the dropped traffic to ICMP echo requests (pings) and/or replies, use the **policy condition icmptype** to specify the appropriate condition. For example,

```
-> policy condition echo icmptype 8
-> policy condition reply icmptype 0
```

802.1p and ToS/DSCP Marking and Mapping

802.1p values can be mapped to different 802.lp values on an individual basis or by using a map group. In addition, ToS or DSCP values can be mapped to 802.1p on a case-by-case basis or via a map group. (Note that any other mapping combination is not supported.)

Marking is accomplished with the following commands:

policy action 802.1p
policy action tos
policy action dscp

Mapping is accomplished through the following commands:

policy map group
policy action map

Note the following:

- Priority for the flow is based on the policy action. The value specified for 802.1p, ToS, DSCP, or the map group determines how the flow is queued.
- The port on which the flow arrives (the ingress port) must be a trusted port. For more information about trusted ports, see "Configuring Trusted Ports" on page 28-9.

In this example, a policy rule (**marking**) is set up to mark flows from 10.10.3.0 with an 802.1p value of 5:

```
-> policy condition my_condition source ip 10.10.3.0 mask 255.255.255.0
-> policy action my_action 802.1p 5
-> policy rule marking condition my_condition action my_action
```

In the next example, the **policy map group** command specifies a group of values that must be mapped; the **policy action map** command specifies what must be mapped (802.1p to 802.1p, ToS/DSCP to 802.1p) and the mapping group that must be used. For more details about creating map groups, see "Creating Map

Groups" on page 28-62.

Here, traffic from two different subnets must be mapped to 802.1p values in a network called Network C. A map group (**tosGroup**) is created with mapping values.

```
-> policy map group tos_group 1-4:4 5-7:7
-> policy condition SubnetA source ip 10.10.5.0 mask 255.255.255.0
-> policy condition SubnetB source ip 12.12.2.0 mask 255.255.255.0
-> policy action map_action map tos to 802.1p using tos_group
```

The **map_action** specifies that ToS values are mapped to 802.1p with the values specified in **tos_group**. With these conditions and action set up, two policy rules can be configured for mapping Subnet A and Subnet B to the ToS network:

```
-> policy rule RuleA condition SubnetA action map_action
-> policy rule RuleB condition SubnetB action map_action
```

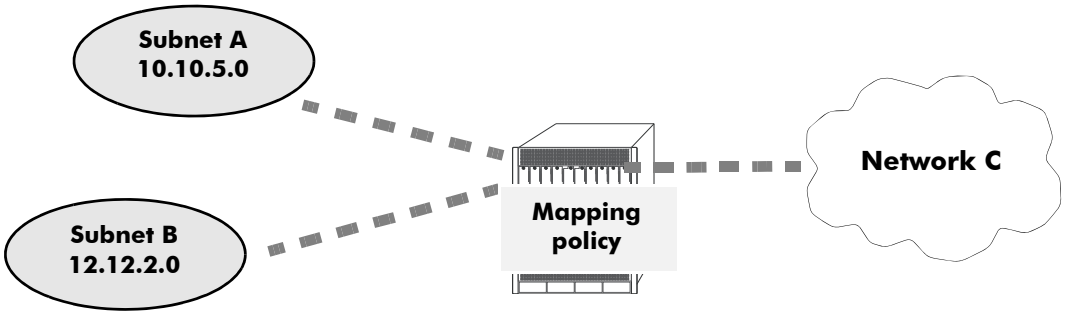


Figure 28-5 : Mapping Application

Policy Based Routing

Policy Based Routing (PBR) allows a network administrator to define QoS policies that override the normal routing mechanism for traffic matching the policy condition.

Note. When a PBR QoS rule is applied to the configuration, it is applied to the entire switch, unless you specify a built-in port group in the policy condition.

Policy Based Routing can be used to redirect traffic to a particular gateway based on source or destination IP address, source or destination network group, source or destination TCP/UDP port, a service or service group, IP protocol, or built-in source port group.

Traffic can be redirected to a particular gateway regardless of what routes are listed in the routing table. Note that the gateway address does not have to be on a directly connected VLAN; the address can be on any network that is learned by the switch.

Note. If the routing table has a default route of 0.0.0.0, traffic matching a PBR policy is redirected to the route specified in the policy. For information about viewing the routing table, see Chapter 17, "Configuring IP."

Policy Based Routing can be used to redirect untrusted traffic to a firewall. In this case, note that reply packets are not allowed back through the firewall.

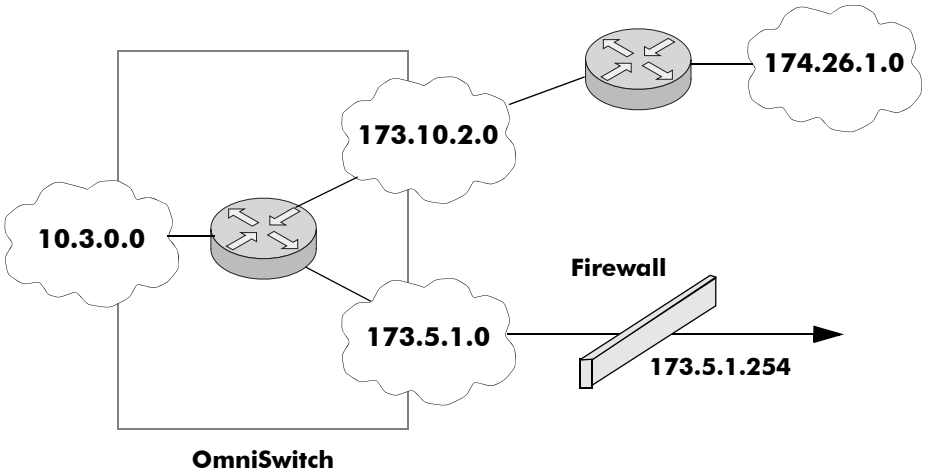


Figure 28-6 : Routing All IP Source Traffic through a Firewall

In this example, all traffic originating in the 10.3 network is routed through the firewall, regardless of whether or not a route exists.

```
-> policy condition Traffic3 source ip 10.3.0.0 mask 255.255.0.0
-> policy action Firewall permanent gateway ip 173.5.1.254
-> policy rule Redirect_All condition Traffic3 action Firewall
```

Note that the functionality of the firewall is important. In the example, the firewall is sending the traffic to be routed remotely. If you instead set up a firewall to send the traffic back to the switch to be routed, you must set up the policy condition with a built-in source port group so that traffic coming back from the firewall does not get looped and sent back out to the firewall.

For example:

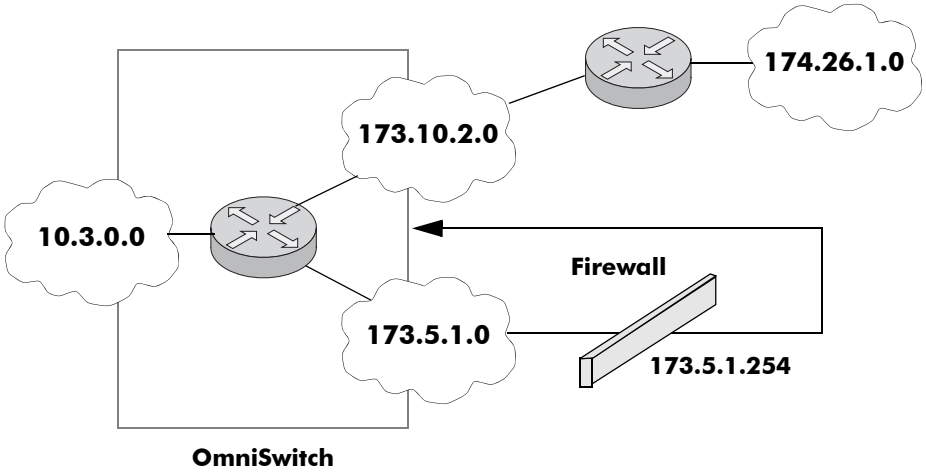


Figure 28-7 : Using a Built-In Port Group

In this scenario, traffic from the firewall is sent back to the switch to be re-routed. But because the traffic re-enters the switch through a port that is not in the Slot01 port group, the traffic does not match the Redirect_All policy and is routed normally through the switch.

```
-> policy condition Traffic3 source ip 10.3.0.0 mask 255.255.0.0 source port
group Slot01
-> policy action Firewall permanent gateway ip 173.5.1.254
-> policy rule Redirect_All condition Traffic3 action Firewall
```

Make sure to enter the **qos apply** command to activate the policy rule on the switch. Otherwise the rule is saved as part of the pending configuration, but is not active.

Non-Contiguous Masks

Non-contiguous masks expand the accepted inputs for the Access Control List (ACL) netmask to facilitate load distribution through Policy Based Routing (PBR). The feature allows masks consisting of any combination of zeros (0) and ones (1). Previously only traditional netmasks were supported and only allowed up to eight bits of zeros to be sparsely distributed in the mask. Traditional netmasks begin with ones followed by a contiguous sequence of zeros (for example, 255.255.255.0). The non-contiguous mask feature supports IPv4 and IPv6 address masks in policy condition statements that contain any sequence of zeros and ones.

The following example illustrates how ACLs can be used to select a subset of the source IP address to be matched and then routed to various gateway-IP addresses using conditions, actions, and rules. The next-hop gateway-IP address should be on a subnet that the router has a directly connected interface for.

Non-contiguous mask examples

A network administrator wishes to distribute IPv4 traffic from the 12.0.0.0 network to a group of servers. In this example there are eight servers that can perform the requested service and the traffic can be distributed depending on the source IP address. These servers reside at addresses 10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4, 10.0.0.5, 10.0.0.6, 10.0.0.7 and 10.0.0.8.

The policy condition commands define a condition that will match one of eight large sets of source IPv4 addresses. The zeros in the mask define don't care or any value matches. The ones in the mask define the care bits that must match the portion of the address defined by the source IP portion of the command. The first condition command matches the source IP address set described as follows:

- 12.any.any.0
- 12.any.any.8
- 12.any.any.16
- 12.any.any.(0+(n*8))

The policy action commands direct the set of source addresses to a specific IP address. The policy rule commands combine the condition and action to form the specific behavior.

```
-> policy condition c1 source ip 12.0.0.0 mask 255.0.0.7
-> policy action a1 permanent gateway-ip 10.0.0.1
-> policy rule r1 condition c1 action a1

! route 1,9,17,33,(1+(n*8))
-> policy condition c2 source ip 12.0.0.1 mask 255.0.0.7
-> policy action a2 permanent gateway-ip 10.0.0.2
-> policy rule r2 condition c2 action a2

! route 2,10,18,34,(2+(n*8))
-> policy condition c3 source ip 12.0.0.2 mask 255.0.0.7
-> policy action a3 permanent gateway-ip 10.0.0.3
-> policy rule r3 condition c3 action a3
```

```
! route 3,11,19,35,(3+(n*8))
-> policy condition c4 source ip 12.0.0.3 mask 255.0.0.7
-> policy action a4 permanent gateway-ip 10.0.0.4
-> policy rule r4 condition c4 action a4

! route 4,12,20,36,(4+(n*8))
-> policy condition c5 source ip 12.0.0.4 mask 255.0.0.7
-> policy action a5 permanent gateway-ip 10.0.0.5
-> policy rule r5 condition c5 action a5

! route 5,13,21,37,(5+(n*8))
-> policy condition c6 source ip 12.0.0.5 mask 255.0.0.7
-> policy action a6 permanent gateway-ip 10.0.0.6
-> policy rule r6 condition c6 action a6

! route 6,14,22,38,(6+(n*8))
-> policy condition c7 source ip 12.0.0.6 mask 255.0.0.7
-> policy action a7 permanent gateway-ip 10.0.0.7
-> policy rule r7 condition c7 action a7

! route 7,15,23,39,(7+(n*8))
-> policy condition c8 source ip 12.0.0.7 mask 255.0.0.7
-> policy action a8 permanent gateway-ip 10.0.0.8
-> policy rule r8 condition c8 action a8
-> qos apply
```

Note the following regarding the use and configuration of IPv4 non-contiguous masks.

- Automatic resolution via Address Resolution Protocol (ARP) for next-hop (permanent gateway-IP) addresses is not supported if a mask contains more than 8-bits of non-contiguous zeros. As a result, other mechanisms have to be used to resolve the MAC addresses such as server load balancing ping probing or static ARP entries.
- A Server Load Balancing (SLB) configuration can be used to probe IPv4 addresses. This allows for dynamic resolution of the IPv4 next hop policy based route. For example:

```
-> vlan 14 admin-state enable
-> vlan 14 members port 1/14 untagged
-> ip interface "v4_v14" vlan 14 admin-state enable
-> ip address 10.0.0.254
-> ip slb cluster pbr_servers vip 1.2.3.4
-> ip slb server ip 10.0.0.1 cluster pbr_servers
-> ip slb server ip 10.0.0.2 cluster pbr_servers
-> ip slb server ip 10.0.0.3 cluster pbr_servers
-> ip slb server ip 10.0.0.3 cluster pbr_servers
-> ip slb server ip 10.0.0.4 cluster pbr_servers
-> ip slb server ip 10.0.0.5 cluster pbr_servers
-> ip slb server ip 10.0.0.6 cluster pbr_servers
-> ip slb server ip 10.0.0.7 cluster pbr_servers

-> ip slb cluster pbr_servers ping period 1
-> ip slb cluster pbr_servers ping timeout 1000

IPv6 example using an IPv6 gateway address:

-> policy condition c9 source ipv6 2000::1 mask e000::7
-> policy action a9 permanent gateway-ipv6 2607:f0d0:2001:000a:0000:0000:0010
-> policy rule r9 condition c9 action a9
-> qos apply
```

Enhanced Congestion Notification (ECN)

Enhanced Congestion Notification (ECN) is an improved mechanism for responsive flows, such as TCP/UDP, to proactively detect congestion in the network devices and throttle sender rate of transmission to allow the congestion point to clear up, instead of ignoring the condition resulting into packet drops due to complete resource exhaustion as seen in regular TCP.

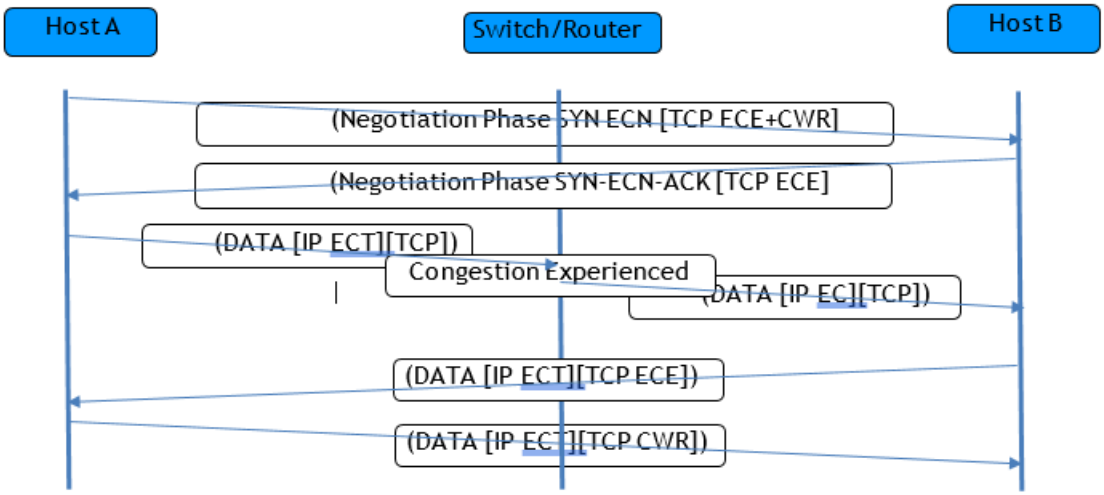
ECN Mechanism

ECN detects and notifies congestion between two endpoints in TCP/IP and UDP/IP based networks. ECN usage requires that it is supported and enabled by both endpoints.

For ECN to work properly, both IP (for marking congestion) and TCP/UDP (for negotiating and responding to congestion) layers must support this.

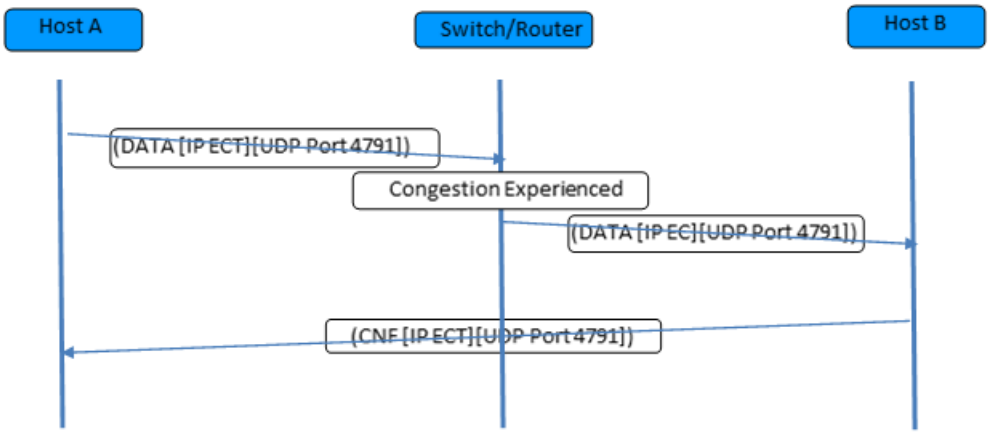
ECN Mechanism Using TCP

The end devices (Servers/PC) first negotiate that ECN is supported on both ends using TCP SYN before setting the packets with ECN code (ECT0 - ECN Capable Transport, or ECT1 - ECN Capable Transport). The network devices also need to be configured to remark the packets with ECN code CE (Congestion Encountered) when congestion is experienced. Network devices such as switches/routers use some form of Random Early Drop queueing technique for detecting congestion and remarking the packet with CE when the packet exits device. When the end receiver detects the ECN code is set to CE, it informs the end transmitter of this congestion in the network by sending a reply with TCP ECN flag set to ECE. This signals the transmitter to reduce its rate of transmission for some time as well as acknowledges back to receiver with TCP flag CWR indicating the CE message was received.



ECN Mechanism for ROCEv2 Support Using UDP

RDMA over Converged Ethernet (RoCEv2) uses IP Header ECN marking mechanism for congestion notification over UDP data gram for InfiniBand Header and data encapsulation. UDP Destination Port 4791 is reserved for RoCEv2. This mechanism is same as ECN marking for TCP packets where the receiving end device send back a congestion notification to the source end device if the network devices in between detected congestion and marked the packets. Upon receiving congestion notification, the sender lowers transmit rate to allow congestion point to clear.



Enhanced Congestion Notification (ECN) Profile Configuration

This section describes the Enhanced Congestion Notification profile (ECNP) configuration commands.

Creating a Custom ECN Profile

A custom profile is created by importing a predefined ECN profile to use as a template for the custom profile. Only ECNP profile ID: 1, name: ecnp-1 can be used to create a custom profile.

ECN is supported only for IP unicast traffic flows.

Use **qos ecnp import ecnp** command to create a custom ECN profile in a non-Data Center Bridging (NDCB) mode.

```
-> qos ecnp ecnptest import ecnp 1
-> qos ecnp 3 import ecnp-1
```

Use the **no** form of the command to remove an existing custom ECN profile from the switch configuration.

```
-> no qos encp 3
```

Configure ECNP Threshold Value

Use **qos ecnp value** command configure the minimum and maximum queue threshold value of a custom ECN profile.

Minimum threshold value cannot be greater the configured maximum threshold value.

```
-> qos ecnp ecnp6 min-threshold 20 max-threshold 75
```

Attach ECNP to a Custom QSP

Use **qos qsp ecn-profile** command to attach a custom ECNP to a custom QSet profile (QSP). The valid custom profile ID range is 3–8.

```
-> qos qsp qsptest ecn-profile encptest
-> qos qsp 6 ecn-profile 3
```

Enable ECN Functionality for a Queue Instance

Use **qos qsi** command to enable or disable ECN functionality for the queue of a port or linkagg.

When enabled, marking of ECT congested packets on the selected egress queue is enabled on the configured port only. ECN can only be activated for lower priority queues 5-8.

```
-> qos qsi port 1/1/31 qp 5 ecn admin-state enable
-> qos qsi port 1/1/31 qp 5 ecn admin-state disable
```

29 Managing Policy Servers

Quality of Service (QoS) policies that are configured through the PolicyView network management application are stored on a Lightweight Directory Access Protocol (LDAP) server. PolicyView is an OmniVista application that runs on an attached workstation.

In This Chapter

This chapter describes how LDAP directory servers are used with the switch for policy management. There is no required configuration on the switch. When policies are created on the directory server through PolicyView, the PolicyView application automatically configures the switch to communicate with the server. This chapter includes information about modifying configuration parameters through the Command Line Interface (CLI) if manual reconfiguration is necessary. For more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Throughout this chapter the term *policy server* is used to refer to LDAP directory servers used to store policies. Procedures described in this chapter include:

- "Installing the LDAP Policy Server" on page 29-3
- "Modifying Policy Servers" on page 29-4
- "Verifying the Policy Server Configuration" on page 29-7

Policy Server Defaults

Defaults for the **policy server** command are as follows:

Description	Keyword	Default
The port number for the server	**port**	389 (SSL disabled) 636 (SSL enabled)
Priority value assigned to a server, used to determine search order	**preference**	0 (lowest)
Whether a Secure Socket Layer is configured for the server	**ssl | no ssl**	no ssl

Policy Server Overview

The Lightweight Directory Access Protocol (LDAP) is a standard directory server protocol. The LDAP policy server client in the switch is based on RFC 2251. Currently, only LDAP servers are supported for policy management.

When the policy server is connected to the switch, the switch is automatically configured to communicate with the server to download and manage policies created by the PolicyView application. There is no required user configuration. (Note that the LDAP policy server is automatically installed when the PolicyView application is installed.)

Note. The switch has separate mechanisms for managing QoS policies stored on an LDAP server and QoS policies configured directly on the switch. For more information about creating policies directly on the switch, see Chapter 28, "Configuring QoS."

Information about installing the LDAP policy server is included in this chapter. Consult the server manufacturer's documentation for detailed information about configuring the server.

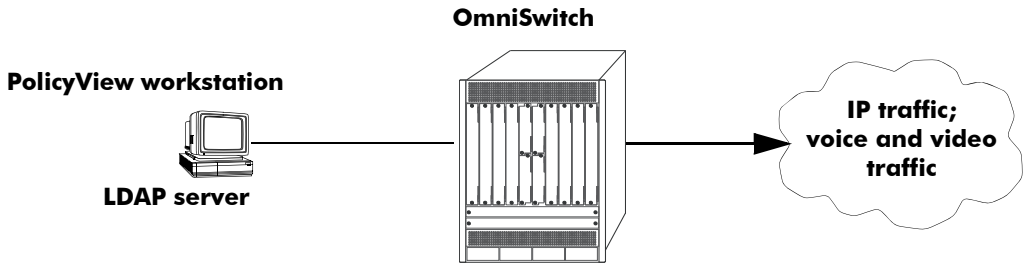


Figure 29-1 : Policy Server Setup

Installing the LDAP Policy Server

Currently Netscape Directory Server 4.15 is supported. The server software is bundled with the PolicyView NMS application.

1. Install the directory server software on the server.
2. Install the Java Runtime Environment on the server.

See your server documentation for additional details on setting up the server.

See the next sections of this chapter for information about modifying policy server parameters or viewing information about policy servers.

Modifying Policy Servers

Policy servers are automatically configured when the server is installed; however, policy server parameters can be modified if necessary.

Note. SSL configuration must be done manually through the **policy server** command.

Modifying LDAP Policy Server Parameters

Use the **policy server** command to modify parameters for an LDAP policy server.

Keywords for the command are listed here:

Policy server keywords	
port	**password**
admin	**searchbase**
preference	**ssl**
user	

For information about policy server parameter defaults, see "Policy Server Defaults" on page 29-2.

Disabling the Policy Server From Downloading Policies

Policy servers can be prevented from downloading policies to the switch. By default, policy servers are enabled to download policies.

To disable a server, use the **policy server** command with the **admin-state** keyword and **disable** option.

```
-> policy server 10.10.2.3 admin-state disable
```

In this example, an LDAP server with an IP address of 10.10.2.3 is not used to download policies. Any policies already downloaded to the switch are not affected by disabling the server.

To re-enable the server, specify **enable**.

```
-> policy server 10.10.2.3 admin-state enable
```

The server is now available for downloading policies.

To delete a policy server from the configuration, use the **no** form of the command with the relevant IP address:

```
-> no policy server 10.10.2.3
```

If the policy server is not created on the default port, the **no** form of the command must include the port number. For example:

```
-> no policy server 10.10.2.4 5000
```

Modifying the Port Number

To modify the port, enter the **policy server** command with the **port** keyword and the relevant port number.

```
-> policy server 10.10.2.3 port 5000
```

Note that the port number must match the port number configured on the policy server.

If the port number is modified, any existing entry for that policy server is not removed. Another entry is simply added to the policy server table.

Note. If you enable SSL, the port number is automatically set to 636. (This does not create another entry in the port table.)

For example, if you configure a policy server with port 389 (the default), and then configure another policy server with the same IP address but port number 5000, two entries display on the **show policy server** screen.

```
-> policy server 10.10.2.3
-> policy server 10.10.2.3 port number 5000
-> show policy server

Server IP Address     port   enabled  status   primary
------+--------------+-----+---------+--------+--------
1      10.10.2.3       389   Yes       Up       X
2      10.10.2.3      5000   No        Down     -
```

To remove an entry, use the **no** form of the **policy server** command. For example:

```
-> no policy server 10.10.2.3 port number 389
```

The first entry is removed from the policy server table.

Modifying the Policy Server Username and Password

A user name and password can be specified so that only specific users can access the policy server.

```
-> policy server 10.10.2.3 user kandinsky password blue
```

If this command is entered, a user with a username of **kandinsky** and a password of **blue** is able to access the LDAP server to modify parameters on the server itself.

Modifying the Searchbase

The searchbase name is “o=alcatel.com” by default. To modify the searchbase name, enter the **policy server** command with the **searchbase** keyword. For example:

```
-> policy server 10.10.2.3 searchbase "ou=qo,o=company,c=us"
```

Note that the searchbase path must be a valid path in the server directory structure.

Configuring a Secure Socket Layer for a Policy Server

A Secure Socket Layer (SSL) can be configured between the policy server and the switch. If SSL is enabled, the PolicyView application can no longer write policies to the LDAP directory server.

By default, SSL is disabled. To enable SSL, use the **policy server** command with the **ssl** option. For example:

```
-> policy server 10.10.2.3 ssl
```

SSL is now enabled between the specified server and the switch. The port number in the switch configuration is automatically set to 636, which is the port number typically used for SSL; however, the port number must be configured with whatever port number is set on the server. For information about configuring the port number, see "Modifying the Port Number" on page 29-5.

To disable SSL, use **no ssl** with the command:

```
-> policy server 10.10.2.3 no ssl
```

SSL is disabled for the 10.10.2.3 policy server. No additional policies can be saved to the directory server from the PolicyView application.

Loading Policies From an LDAP Server

To download policies (or rules) from an LDAP server to the switch, use the **policy server load** command. Before a server can download policies, it must also be set up and operational (able to bind).

To download policies from the server, enter the following:

```
-> policy server load
```

Use the **show policy server long** command to display the last load time. For example:

```
-> show policy server long
LDAP server 0
  IP address: 10.10.2.3,
  TCP port: 16652,
  Enabled: Yes,
  Operational Status: Down,
  Preference: 99,
  Authentication: password,
  SSL: Disabled,
  login DN: cn=DirMgr
  searchbase: o=company
  Last load time: 02/14/02 16:38:18
```

Removing LDAP Policies From the Switch

To flush LDAP policies from the switch, use the **policy server flush** command. Note that any policies configured directly on the switch through the CLI *are not affected* by this command.

```
-> policy server flush
```

Interaction With CLI Policies

Policies configured via PolicyView can only be modified through PolicyView. They cannot be modified through the CLI. Any policy management done through the CLI only affects policies configured through the CLI. For example, the **qos flush** command only removes CLI policies; LDAP policies are not affected.

Also, the **policy server flush** command removes only LDAP policies; CLI policies are not affected.

Note. If polices are applied from PolicyView or vice versa, it activates all current configuration.

For more information about configuring policies through the CLI, see Chapter 28, "Configuring QoS."

Verifying the Policy Server Configuration

To display information about authentication and policy servers, use the following commands:

show policy server	Displays information about servers from which policies can be downloaded to the switch.
show policy server long	Displays detailed information about an LDAP policy server.
show policy server statistics	Displays statistics about policy directory servers.
show policy server rules	Displays the names of policies originating on a directory server that have been downloaded to the switch.
show policy server events	Displays any events related to a directory server.

30 Configuring Access Guardian

Access Guardian refers to the following OmniSwitch security functions that work together to provide a dynamic, proactive network security solution:

- **Universal Network Profile (UNP)**—Access Guardian is configured and applied through the framework of the UNP feature. UNP is enabled on switch ports to activate Access Guardian functionality that is used to authenticate and classify users into UNP profiles. Each profile is mapped to a VLAN ID or Service Access Point (SAP) to which the user is dynamically assigned. Specific UNP port configurations help to simplify and easily replicate the same configuration across multiple ports.
- **Authentication, Authorization, and Accounting (AAA)**—Provides the switch-based authentication and accounting configuration that defines the RADIUS-capable servers to use for each type of Access Guardian authentication (802.1X, MAC, and Captive Portal). AAA profiles define a specific AAA configuration that can be applied at the port level (overrides the global AAA configuration).
- **Bring Your Own Device (BYOD) - OmniSwitch / UPAM or ClearPass Integration**: The OmniSwitch leverages Access Guardian functionality along with the OmniVista Unified Policy Access Manager (UPAM) or the ClearPass Policy Manager (CPPM) to provide the overall BYOD solution. BYOD allows a wired guest, device, or authenticated user to connect to the network through an OmniSwitch edge device using the UPAM or CPPM for unified authentication. UPAM and CPPM provide the framework for device onboarding, guest registration, and authentication, as well as device posture checking and profiling.
- **Captive Portal**—Internal and external Captive Portal Web-based authentication. Internal Captive Portal authentication is provided through an internal Web server on the OmniSwitch that presents default or customized Web pages to the user. A post-authentication and/or post-classification process to validate user credentials and dynamically assign a new role (policy list) to enforce user access to the network. External guest Captive Portal authentication is provided through the OmniSwitch Access Guardian interaction with the OmniVista Unified Policy Access Manager or the ClearPass Policy Manager.
- **Quarantine Manager and Remediation (QMR)**—QMR is a switch-based application that restricts the network access of known quarantined users and provides a remediation path to allow quarantined users to regain their network access.
- **IoT Device Profiling IoT**—Device Profiling allows the network administrators to support and manage smart phones, Tablets and other devices connecting to the network. The IoT Device Profiling uses DHCP FingerPrinting and MAC OUI (MAC Vendors) to identify IoT devices.

In This Chapter

This chapter provides an overview of Access Guardian security features and describes how to configure these features through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following information and procedures are included in this chapter:

- "Access Guardian Defaults" on page 30-3.
- "Quick Steps for Configuring Access Guardian" on page 30-10
- "Access Guardian Overview" on page 30-12.
- "Interaction With Other Features" on page 30-27.
- "Configuring Port-Based Network Access Control" on page 30-33.
- "Configuring UNP Profiles" on page 30-58.
- "Configuring UNP Classification Rules" on page 30-80.
- "Using Router Domain Authentication" on page 30-84
- "Using Captive Portal Authentication" on page 30-91.
- "OmniAccess Stellar AP Integration" on page 30-101.
- "Using L2 GRE Tunneling" on page 30-113.
- "Using Quarantine Manager and Remediation" on page 30-128.
- "Access Guardian Application Examples" on page 30-130.
- "Verifying Access Guardian Users" on page 30-144.
- "Verifying the Access Guardian Configuration" on page 30-149.
- "Bring Your Own Devices (BYOD) Overview" on page 30-150.
- "Multicast Domain Name System" on page 30-162.
- "Simple Service Discovery Protocol" on page 30-163.
- "Zero Configuration Networking (mDNS and SSDP)" on page 30-167.
- "BYOD Application Examples" on page 30-180.
- "IoT Device Profiling" on page 30-197.

For more information about configuring the UNP feature, see Chapter 40, "Access Guardian Commands,"

Access Guardian Defaults

This sections contains the default configuration settings for the Access Guardian security functions that are implemented through the Universal Network Profile (UNP), Captive Portal, Quarantine Manager and Remediation (QMR) features.

Access Guardian Global Configuration Defaults

The following global default values are applied to traffic received on all UNP ports or link aggregates.

Description	Keyword	Default
Authentication server down UNP	**unp auth-server-down**	None
Authentication server down timer	**unp auth-server-down-timeout**	60 seconds
Port bounce for MAC authenticated (non-supplicant) devices.	**unp redirect port-bounce**	Disabled
The amount of time to filter MAC addresses to trigger authentication	**unp redirect pause-timer**	0 (timer disabled)
HTTP proxy port number	**unp redirect proxy-server-port**	80, 8080, and 443
IP address to which HTTP traffic is redirected	**unp redirect-server**	None
Additional IP addresses to which a user can be redirected, other than the redirect server.	**unp redirect allowed-name**	None
Dynamic VLAN configuration for VLAN profiles.	**unp dynamic-vlan-configuration**	Disabled
Dynamic VLAN profile configuration.	**unp dynamic-profile-configuration**	Disabled
UNP Customer Domain ID	**unp domain description**	All bridge and access ports belong to domain 0.
Configures EAPoL version V1 or V3 globally on the switch for 802.1x authentication of users on UNP port.	**unp 802.1x eapol-version**	EAPoL version V1.

Access Guardian Profile Defaults

Access Guardian profile-based functionality is implemented through the configuration of Universal Network Profiles (UNP). When a UNP profile is created with the **unp profile** command, the following default configuration is defined for the profile:

Description	Command	Default
QoS policy list.	**unp profile qos-policy-list**	No list assigned
Location-based policy	**unp profile location-policy**	No policy assigned
Time-based policy.	**unp profile period-policy**	No policy assigned
Internal Captive Portal authentication.	**unp profile captive-portal-authentication**	Disabled
Captive Portal configuration profile.	**unp profile captive-portal-profile**	No Captive Portal profile assigned
The authentication flag status for successful authentication.	**unp profile authentication-flag**	Disabled
Create a tagged association between a UNP port and the VLAN or service that is mapped to the profile.	**unp profile mobile-tag**	Disabled
Maximum bandwidth value for traffic received on UNP ports assigned to the profile.	**unp profile maximum-ingress-bandwidth**	No limit set
Maximum bandwidth value for traffic sent on UNP ports assigned to the profile.	**unp profile maximum-egress-bandwidth**	No limit set
How much the traffic can burst over the maximum ingress bandwidth rate.	**unp profile maximum-ingress-depth**	Ingress bandwidth value divided by 25 or 2K (if calculation equals 0 or 1)
How much the traffic can burst over the maximum egress bandwidth rate.	**unp profile maximum-egress-depth**	Egress bandwidth value divided by 25 or 2K (if calculation equals 0 or 1)
The amount of time an authenticated device can remain logged after the MAC address for the device has aged out.	**unp profile inactivity-interval**	10 seconds
Service Assurance Agent (SAA) profile.	**unp profile saa-profile**	No SAA profile assigned
Profile mapping.	**unp profile map vlan** **unp profile map service-type spb** **unp profile map service-type vxlan** **unp profile map service-type l2gre** **unp profile map service-type static**	None

See "UNP Profiles" on page 30-16 for more information.

Access Guardian UNP Port Defaults

Access Guardian port-based functionality is implemented through the UNP feature. When UNP is enabled on a switch port or link aggregate with the **unp port-type** command, the following default configuration for UNP ports is applied:

Description	Command	Default
Port bounce for MAC authenticated (non-supplicant) devices.	**unp redirect port-bounce**	Disabled
802.1X authentication	**unp 802.1x-authentication**	Disabled
Alternate UNP profile for 802.1X authenticated traffic.	**unp 802.1x-authentication pass-alternate**	None
Bypass 802.1X authentication for supplicants	**unp 802.1x-authentication bypass-8021x**	Disabled
Attempt MAC authentication or classification when 802.1X authentication fails.	**unp 802.1x-authentication failure-policy**	Classification
Attempt 802.1X authentication after MAC authentication when 802.1X bypass is enabled.	**unp mac-authentication allow-eap**	None
MAC authentication	**unp mac-authentication**	Disabled
Alternate UNP for MAC authenticated traffic.	**unp mac-authentication pass-alternate**	None
Rule-based classification	**unp classification**	Disabled
Trust the VLAN ID of a tagged packet to determine how the packet is classified.	**unp trust-tag**	Disabled
Default UNP profile	**unp default-profile**	None
Domain ID assignment.	**unp domain**	0
AAA configuration profile assignment.	**unp aaa-profile**	None
Port template assignment.	**unp port port-template**	UNP bridge ports: bridgeDefaultPortTemplate UNP access ports: accessDefaultPortTemplate
Allow flooding of egress broadcast, unknown unicast, or multicast traffic (applies only to UNP bridge ports).	**unp direction**	Both (blocked)
The administrative status of the UNP configuration on the port.	**unp admin-state**	Enabled

Description	Command	Default
The type of service (SPB or VXLAN) automatically created based on traffic received on the UNP access port.	**unp dynamic-service**	No service is dynamically created.
The amount of time before an EAP Request Identity is retransmitted.	**unp 802.1x-authentication tx-period**	30 seconds
The amount of time before the switch times out an 802.1X user attempting to authenticate.	**unp 802.1x-authentication supp-timeout**	30 seconds
The maximum number of requests retransmitted before the session times out.	**unp 802.1x-authentication max-req**	2
Layer 2 profile that specifies how control packets are processed on UNP access ports.	**unp l2-profile**	unp-def-access-profile

See "UNP Ports" on page 30-22 for more information.

Access Guardian Global AAA Parameter Defaults

The following default AAA (authentication, authorization, and accounting) parameter settings are applied to Access Guardian device authentication and accounting sessions. (Note that the AAA profile settings override the global settings when the profile is applied to a UNP port.)

Description	Command	Default
The RADIUS server configuration for device authentication.	**aaa device-authentication**	None
The RADIUS server or switch logging configuration for accounting sessions.	**aaa accounting**	None
The RADIUS Calling-Station-Id attribute value	**aaa accounting radius calling-station-id**	User MAC address
The status of automatic 802.1X re-authentication.	**aaa 802.1x re-authentication**	Disabled
The status of the trust RADIUS option for automatic 802.1X re-authentication.	**aaa 802.1x re-authentication trust-radius**	Disabled
The amount of time between interim accounting updates per user session.	**aaa interim-interval**	600 seconds
User session time limit.	**aaa session-timeout**	Timer is disabled

Description	Command	Default
The amount of time before an inactive user is logged out of a session.	**aaa inactivity-logout**	Timer is disabled
The RADIUS NAS-Port attribute value	**aaa radius nas-port-id**	User port (chassis/slot/port)
The RADIUS NAS-Identifier attribute value	**aaa radius nas-identifier**	System name
The source IP address for the NAS-IP-Address attribute	**aaa radius nas-ip-address**	IP address of the interface used to send the RADIUS packet.
The MAC address format to use when RADIUS client attributes specify a MAC address value.	**aaa radius mac-format**	No delimiter Uppercase characters

Access Guardian AAA Profile Defaults

An AAA profile defines and applies specific settings to UNP ports, link aggregates, or an Access Guardian Captive Portal profile. The following table lists the default profile settings that are defined when an AAA profile is created through the **aaa profile** command:

Description	Keyword	Default
The RADIUS server configuration for device authentication.	**device-authentication**	None
The RADIUS server or switch logging configuration for accounting sessions.	**accounting**	None
The RADIUS Calling-Station-Id attribute value.	**accounting radius calling-station-id**	User MAC address
The status of automatic 802.1X re-authentication.	**802.1x re-authentication**	Disabled
The status of the trust RADIUS option for automatic 802.1X re-authentication.	**802.1x re-authentication trust-radius**	Disabled
The amount of time between interim accounting updates per user session.	**interim-interval**	600 seconds
User session time limit.	**session-timeout**	Timer is disabled
The amount of time before an inactive user is logged out of a session.	**inactivity-logout**	Timer is disabled
The RADIUS NAS-Port attribute value.	**radius nas-port-id**	User port (chassis/slot/port)

Description	Keyword	Default
The RADIUS NAS-Identifier attribute value.	**radius nas-identifier**	System name
The RADIUS NAS-IP-Address attribute value.	**radius nas-ip-address**	IP address of the interface used to send the RADIUS packet.
The MAC address format to use when RADIUS client attributes specify a MAC address value.	**radius mac-format**	No delimiter Uppercase characters

Access Guardian Captive Portal Defaults

The following global default configuration settings apply to the OmniSwitch internal implementation of the Captive Portal feature:

Description	Command	Default
Redirect URL name	**captive-portal name**	"captive-portal.com"
Redirect IP address	**captive-portal ip-address**	10.123.0.1
Redirect user after successful Captive Portal login.	**captive-portal success-redirect-url**	No redirect
Proxy server port number	**captive-portal proxy-server-port**	8080
Number of login attempts allowed per Captive Portal session.	**captive-portal retry-count**	3
QoS policy list or UNP profile to apply to a user device after a successful Captive Portal login.	**captive-portal authentication-pass**	None
The amount of time between interim accounting updates for Captive Portal sessions.	**aaa interim-interval**	600 seconds
Captive Portal session time limit.	**aaa session-timeout**	Timer is disabled
The amount of time before an inactive user is logged out of a Captive Portal session.	**aaa inactivity-logout**	Timer is disabled

Access Guardian Captive Portal Profile Defaults

A Captive Portal profile defines and applies specific settings to devices classified into a UNP profile to which the Captive Portal profile is assigned. The following table lists the default profile settings that are

defined when a Captive Portal profile is created through the **captive-portal-profile** command:

Description	Keyword	Default
AAA configuration profile assigned to the Captive Portal profile.	**aaa-profile**	No profile assigned (Global AAA settings apply)
Redirect user after successful login.	**success-redirect-url**	No redirect
Number of login attempts allowed.	**retry-count**	3
QoS policy list to apply after successful Captive Portal login.	**authentication-pass policy-list**	No list assigned
UNP profile to apply after successful Captive Portal login.	**authentication-pass profile**	No profile assigned
Change profile assignment after successful Captive Portal login.	**authentication-pass profile-change**	Disabled

Access Guardian QMR Defaults

The following global default configuration settings apply for the OmniSwitch implementation of the Quarantine Manage and Remediation (QMR) feature:

Description	Command	Default
The URL for a remediation server	**qmr quarantine path**	None
Whether a “Quarantined” page is sent to the user when a remediation server URL is not configured	**qmr quarantine page**	No page is sent
The IP network addresses that a restricted quarantined user is allowed to access	**qmr quarantine allowed-name**	None
Proxy server port number	**qmr quarantine custom-proxy-port**	8080
The name of the Quarantine MAC address group on the switch.	**qos quarantine mac-group**	No name is configured

Quick Steps for Configuring Access Guardian

The following procedure provides a brief tutorial for setting up the OmniSwitch implementation of Access Guardian network access control. For additional configuration tutorials, see "Access Guardian Application Examples" on page 30-130 and "Quick Steps for Configuring Captive Portal Authentication" on page 30-93.

1 Configure the RADIUS server to use for device authentication (802.1X, MAC, or Captive Portal). For example, the following commands define the RADIUS server for MAC device authentication:

```
-> aaa radius-server rad1_mac host 10.135.60.44 hash-key secret retransmit 3
timeout 2 auth-port 1812 acct-port 1813
-> aaa device-authentication mac rad1_mac
```

2 Configure the RADIUS server with the IP address of the OmniSwitch and the same shared secret that was assigned through the AAA RADIUS server configuration in Step 1.

3 Add the user name and password details in the RADIUS server.

4 Enable the MAC authentication session timer to determine the amount of time the user session remains active after a successful login (the default time is set to 12 hours). For example:

```
-> aaa mac session-timeout enable
```

5 Configure a UNP profile to which user devices will be assigned. Profile attribute values are applied to devices that are associated with the profile. For example, the following commands create the "na_employee" profile and assign the QoS policy list "naEmpList" to the profile. QoS policy rules contained in the "naEmpList" list are applied to traffic assigned to the "na_employee" profile.

```
-> unp profile na_employee
-> unp profile na_employee qos-policy-list naEmpList
```

The QoS policy list name specified in the above example must already exist in the switch configuration. See "UNP Profile Attributes" on page 30-18 for more information about assigning a QoS policy list and other configurable options for a UNP profile.

6 Configure an additional UNP profile that will serve as a default profile for UNP port configuration. For example, the following command creates the "def_unp" profile that is configured as a default profile for UNP ports configured in Step 10:

```
-> unp profile def_unp
```

7 Configure a VLAN or service mapping for the profiles created in Step 5 and Step 6. Devices that are assigned to a profile will automatically become members of the VLAN or service that is mapped to the profile. For example, the following commands map VLAN 100 to the "na_employee" profile and VLAN 200 to the "def_unp" profile:

```
-> unp profile map na_employee vlan 100
-> unp profile map def_unp vlan 200
```

See "UNP Profile Mapping" on page 30-17 for more information about assigning a VLAN or service parameters to a UNP profile.

8 Configure UNP classification rules that will identify the device traffic to assign to a specific profile. For example, the following command creates a MAC address range rule for profile "na_employee". Any user device with a source MAC address that falls within the specified range is assigned to the profile.

```
-> unp classification mac-range 08-00-27-00-98-0A 08-00-27-00-98-FF profile1
na_employee
```

There are additional types of classification rules that can also be configured to determine UNP profile assignment. See "UNP Classification Rules" on page 30-24 for more information.

9 Configure UNP functionality and port type (bridge or access) on the ports that will connect user devices to the OmniSwitch. Traffic received on bridge ports can be assigned to profiles mapped to a VLAN (VLAN profiles); traffic received on access ports can be assigned to profiles mapped to a service (service profiles). For example, the following command enables UNP bridge port type functionality:

```
-> unp port 1/1/20 port-type bridge
```

10 Enable authentication (802.1X or MAC) for the UNP port to trigger the authentication process for traffic received on the port. For example, the following command enables MAC authentication on UNP port 1/1/20. Traffic received on port 1/1/20 is authenticated through the RADIUS server defined in Step 1:

```
-> unp port 1/1/1 mac-authentication
```

11 Enable the classification status for the UNP port to trigger the use of classification rules to determine the profile assignment for traffic received on the port. Classification rules are only applied when authentication is not enabled or fails to provide a profile assignment for the traffic. For example, the following command enables classification on port 1/1/20:

```
-> unp port 1/1/1 classification
```

12 Assign a default profile to the UNP port. If traffic from a user device that is connected to a UNP port is not classified into any other UNP profile (authentication and classification rules fail), the device is assigned to the default profile configured for the port. For example, the following command configures "def_unp" as the default profile for UNP bridge port 1/1/20:

```
-> unp port 1/1/20 default-profile def_unp
```

For more information about the configurable UNP port options, see "UNP Ports" on page 30-22 for more information.

Access Guardian Overview

Access Guardian is a combination of authentication, device compliance, and access control functions that provide a *proactive* solution for network security. Implemented through the switch hardware and software, Access Guardian helps administrators:

- Determine who is on the network.
- Check if end users are compliant.
- Direct what end users can access within the network.

As shown in the following diagram, the Access Guardian features work together to provide a dynamic, integrated security framework:

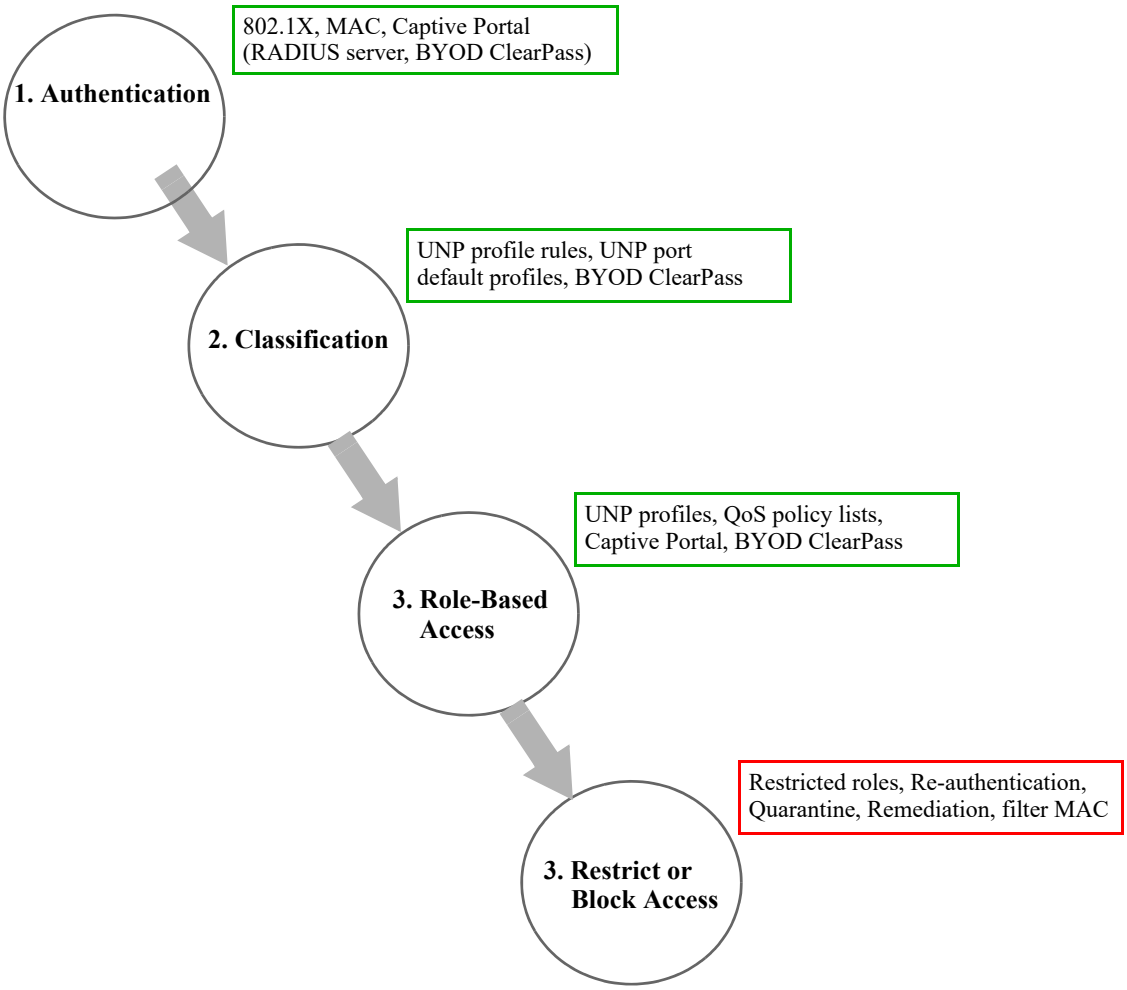


Figure 30-1 : Access Guardian Overview

1 Authentication—Device authentication is attempted through OmniSwitch interaction with a RADIUS server, a Unified Policy Access Manager (UPAM) server, or a ClearPass Policy Manager (CPPM) server. If device authentication fails to return a profile assignment for a device, then device classification is attempted. See "Device Authentication" on page 30-13 for more information.

2 Classification—Device classification into a profile is attempted through the local OmniSwitch configuration or through interaction with a UPAM or CPPM server. See "Device Classification" on

page 30-14 for more information.

3 Role-Based Access—Once a profile assignment is determined for a device through authentication or classification, then the role of the device in the network is determined. The role assigned to a device determines the network resources to which the device is entitled to access. See "Role-based Access" on page 30-15 for more information.

4 Restrict or Block—Steps 1, 2, and 3 of the Access Guardian process may result in a restricted role or even blocking network access for a specific device. Re-authentication and remediation methods are available for such devices.

The Access Guardian feature is implemented through the following switch-based functionality:

- MAC-based and 802.1X-based authentication using a RADIUS-capable server.
- Internal Captive Portal for Web-based authentication. Provides dynamic role change for the user device.
- The Universal Network Profile (UNP) framework to provide network access control and Quality of Service (QoS) on a per-user basis.
- Switch-wide UNP classification rules to classify users based on port and device attributes (for example, source MAC, domain ID, IP address). No authentication required.
- Default UNP classification for traffic not classified through other methods.
- Integration with the Unified Policy Access Manager (UPAM) or the ClearPass Policy Manager (CPPM) as part of the OmniSwitch Bring Your Own Device (BYOD) network access solution.

This chapter documents the functionality of the Access Guardian feature and how it is configured on the OmniSwitch.

Device Authentication

Physical devices attached to a LAN port on the switch through a point-to-point LAN connection can be authenticated through the switch using port-based network access control. This control is available through the Universal Network Profile (UNP) feature implemented on the switch.

Access Guardian uses the UNP feature to provide configurable authentication and classification mechanisms for both 802.1X clients (supplicants) and non-802.1X clients (non-supplicants). The following options for authentication are available:

- **802.1X authentication for supplicants.**

  Uses Extensible Authentication Protocol (EAP) between an end device and a network device (NAS) to authenticate the supplicant through a RADIUS server. If authentication returns a UNP, the supplicant is assigned to that UNP. If a UNP name is not returned or authentication fails, then the UNP port and classification rule configuration provides the network access control for the supplicant.

- **MAC-based authentication for non-supplicants**.

  MAC-based authentication does not require any agent or special protocol on the non-supplicant device; the source MAC address of the device is verified through a RADIUS server. The switch sends RADIUS frames to the server with the source MAC address embedded in the username and password attributes. If authentication returns a UNP name, the non-supplicant is assigned to that profile. If a UNP name is not returned or authentication fails, then the UNP port and classification rule configuration provides the network access control for the non-supplicant.

For non-supplicant authentication, the client MAC address is sent as the username and password. The administrator can configure the password and username on the authentication server as the MAC address of the client. The calling-station-ID, accounting-session-ID are also sent for authentication. All of these IDs can be in uppercase or lowercase.

- **Internal Captive Portal authentication**.

  Internal Captive Portal authentication is a configurable option for a UNP profile that is applied after a user is initially assigned to that profile (after the initial 802.1X or MAC authentication or classification process). Captive Portal provides a secondary level of authentication that is used to apply a new role (QoS policy list) to the user. This type of authentication may change the profile assignment for the user device.

  When a user is classified into a profile that has the Captive Portal option enabled, a Web page is presented to the user device to prompt the user to enter login credentials. The credentials are then authenticated through a RADIUS server. If the authentication process results in a new policy list or new profile, that policy list or profile is applied to the user device. If a policy list or profile is not assigned or authentication fails, the policy list associated with the initial profile is used to define the network access role for the user.

- **External Captive Portal authentication**.

  External Captive Portal authentication is provided through the OmniSwitch Bring Your Own Device (BYOD) solution. Access Guardian, through the UNP port and profile framework, redirects user device traffic to the Unified Policy Access Manager (UPAM) server or the ClearPass Policy Manager (CPPM) server for Guest Access using the UPAM or CPPM Guest module.

802.1X and MAC authentication are Layer 2 mechanisms that are configured and invoked at the port level. A UNP port is enabled with either 802.1X, MAC, or both types of authentication. Devices connected to UNP ports undergo the type of authentication configured on the port.

Internal and external Captive Portal authentication are Layer 3 mechanisms that are invoked through the UNP profile configuration. Devices connected to UNP ports initially undergo Layer 2 authentication and/or classification at the port level to determine an initial UNP profile assignment. Then, based on the profile settings, the user may be redirected for Layer 3 authentication.

The authentication functionality provided allows the administrator to assign the appropriate method of authentication. Multiple authentication methods for multiple users (many users or different types of users, such as IP phones) are supported on the same port.

Device Classification

Successful device authentication can result in a UNP profile assignment for the user device. However, if authentication is not available or does not return a profile name for whatever reason, the following additional UNP device classification methods are available to determine the profile assignment for the user device:

- **UNP classification rules**. Switch-wide classification rules to classify users based on port and device attributes (for example, source MAC, domain ID, IP address). Classification rules are associated with profiles and are applied to traffic received on UNP-enabled ports. When any of the traffic matches one of the classification rules, the user device is dynamically assigned to the matching profile.

- **Alternate pass UNP**. A UNP associated with a UNP port to which traffic is assigned when successful 802.1X or MAC authentication does not return a UNP name.

- **Default UNP**. A UNP associated with a UNP port to which traffic is assigned when other authentication or classification attempts fail to provide a profile name.
- **Trust VLAN tag**. Configured on a UNP port to specify whether or not to trust the VLAN tag of the packets received on the port. If this option is enabled and the VLAN tag matches an existing VLAN in the switch configuration, the traffic is assigned to that VLAN when other authentication or classification attempts fail to provide a profile name.
- **Authentication server down UNP**. A global UNP that provides a temporary profile for devices unable to authenticate because the RADIUS server is unreachable. This profile is associated with a timer that determines how long the device remains in the temporary profile before authentication is attempted again.

Enabling 802.1X and/or MAC authentication on UNP ports is optional; an administrator may decide to use UNP classification rules instead. When enabled, however, the authentication method takes precedence over classification methods.

Role-based Access

When a user is authenticated and/or classified into a UNP profile, the initial role of that user is determined by whether or not there is a QoS policy list associated with the profile.

- If there is no policy list available, then the user has full access to the switch and network resources as provided through the profile VLAN or service domain to which the user was assigned.
- If a policy list is available, then the QoS policy rules associated with that list are applied to the port and traffic of the user device.

Access Guardian provides the following post-authentication and post-classification mechanisms for dynamically changing the role (QoS policy list) applied to a user device.

- **Internal Captive Portal**. User undergoes a secondary authentication process through Captive Portal Web-based authentication. Successful Captive Portal authentication applies the QoS policy list returned from the RADIUS server or specified in the Captive Portal authentication pass configuration. The newly obtained policy list overrides the policy list associated with the profile to which the device was initially assigned. The outcome of this process may also change the profile assignment for the user device. See "Using Captive Portal Authentication" on page 30-91 for more information.
- **Location and Time Policies**. When a user classified into a UNP profile violates a location-based or time-based policy that is associated with the profile, a built-in unauthorized restricted role is applied to that user. The restricted role overrides the policy list associated with the profile.
- **Built-in Restricted Roles**. When one of the built-in restricted roles is applied to a user device, an implicit QoS policy list associated with that role is applied to that device instead of the UNP profile policy list. A custom policy list can be associated with a restricted role to override the built-in role.
- **User-defined Roles.** When the state of a device matches specific conditions configured for a user-defined role, an explicit QoS policy list that is associated with this type of role is applied to the device instead of the UNP profile policy list.

Built-in Restricted Roles

The following types of built-in roles are applied to the user device based on the state of the Access Guardian user:

- **Internal Captive Portal pre-login role**—applied when a user is classified into a UNP profile that has the Captive Portal flag enabled. While in this pre-login state, only DHCP, DNS, ARP, and ICMP traffic from the user device is allowed. In addition, HTTP/HTTPS traffic is trapped and redirected to the internal Captive Portal server.
- **Unauthorized role**—applied when a user classified into a UNP profile violates the location or time policies configured for that profile. Traffic from "unauthorized" users is blocked.
- **QMR role**—applied to quarantined MAC addresses. Traffic from quarantined devices is blocked and HTTP traffic is trapped. When the user opens a browser, HTTP/HTTPS traffic is redirected to a remediation server, if one is configured for QMR on the switch.

Explicit QoS Policy Lists for Built-in Roles

When an Access Guardian user is placed into one of the built-in restricted roles (unauthorized, Captive Portal pre-login, or QMR), the QoS policy list associated with that role is applied to the user. However, it is possible to define and apply an explicit (custom) policy list to a built-in restricted role. When this is done, the explicit policy list will determine how traffic from the user is controlled.

User-Defined Roles

A user-defined role applies an explicit QoS policy list to an Access Guardian user based on the following conditions:

- The user was classified into a specific UNP profile.
- The type of authentication applied to the user device (802.1X, MAC, or none). Can also define this condition based on whether or not the user failed 802.1X or MAC authentication.
- The user is in a Captive Portal post-login state.

The explicit policy list is not applied to a user unless all of the conditions configured for the user-defined role are met.

In addition to these conditions, a precedence value is configured for user-defined roles. This value is used to determine precedence among other user-defined roles. Every time the user context changes for a device, all the user-defined roles are checked to see if there is a role that matches the current user context.

UNP Profiles

Access Guardian role-based network access is achieved through the OmniSwitch Universal Network Profile (UNP) feature. A UNP profile defines network access for one or more user devices. Each device that is assigned to a specific profile is granted network access based on the profile criteria, instead of on an individual MAC address, IP address, or port basis.

Assigning users to a profile provides greater flexibility and scalability across the network. Administrators can use profiles to group users according to function. All users assigned to the same UNP become members of that profile group. The UNP then determines what network resources are available to a group of users, regardless of source subnet, VLAN, or other characteristics.

Dynamic assignment of devices to UNP profiles is achieved through UNP port-based functionality that provides the ability to authenticate and classify device traffic. Device authentication verifies the device identity and provides a UNP name. In the event authentication is not available or is unsuccessful, the following steps are triggered to determine the profile assignment:

1 UNP classification rules are examined to determine if any of the rules match the device traffic. If so, the device is assigned to the profile associated with the matching rule.

2 If there are no matching UNP classification rules, the UNP port-level configuration is used to determine a profile assignment for the device. For example, is there a default UNP profile assigned to the port. If so, the device is assigned to that profile.

UNP Profile Mapping

The mapping of a VLAN ID or service-based parameters determines whether a VLAN-port association (VPA) or a service virtual port association is dynamically created for UNP port traffic that is assigned to the profile. UNP profiles that are mapped to a VLAN ID are referred to as VLAN profiles; UNP profiles that are mapped to service-based parameters are referred to as service profiles.

- **VLAN profile mapping**. This type of profile mapping dynamically creates a VLAN-port association (VPA) for device traffic that is classified into the profile. The VPA represents an association between the UNP bridge port on which the device traffic is received and the VLAN ID mapped to the profile. Once classified into a specific VLAN profile, device traffic is tagged to forward on the UNP VLAN.
- **Service profile mapping**. This type of profile mapping specifies service-based parameter values that are used to dynamically create a Service Access Point (SAP). The SAP becomes a virtual port that is associated with the profile. Once classified into a specific service profile, device traffic is mapped to the SAP and forwarded on the service associated with the SAP. There are two types of service-mappings supported: Shortest Path Bridging (SPB) and Virtual eXtensible LAN (VXLAN).

The OmniSwitch supports two separate traffic domains: VLAN and service. The availability of two types of profile mapping (VLAN and service) provides an efficient method for network access control and dynamic assignment of device traffic into one of these domains.

- An administrator can use VLAN profiles to implement the same UNP name across the entire network infrastructure. Each UNP name can have a different VLAN ID mapping on each switch, as the VLAN mapping configuration applies only to the local switch. For example, the administrator can deploy a UNP named “Engineering” in one building using VLAN 10, while the same UNP deployed in another building can use VLAN 20. The same UNP access controls are applied to all profile devices in each building even though the devices belong to different VLANs.
- A service profile is particularly useful in the OmniSwitch Data Center solution to facilitate virtual machine (VM) discovery and movement. UNP service profiles used for such purposes are also referred to as Virtual Network Profiles (vNPs).

UNP VLANs

When a VLAN is mapped to a UNP profile, specifying a VLAN ID is required. Traffic that is classified with the UNP is assigned to the associated VLAN. There are two methods for creating this type of VLAN:

- Using standard VLAN management commands, create the VLAN then assign the VLAN to the UNP at the time the profile mapping is configured.
- Enabling the UNP dynamic VLAN configuration option to automatically create the VLAN, if it does not exist, at the time the UNP profile mapping is configured.

VLANs that are automatically created at the time the profile mapping is configured are referred to as UNP dynamic VLANs. These VLANs carry many of the same attributes as standard VLANs, such as:

- The VLAN status (enabled or disabled) is configurable.
- Additional ports (tagged and untagged) can be assigned to dynamic VLANs.

- The STP status is configurable and is enabled by default for dynamic VLANs. This STP instance is included in the maximum number of 1x1 STP instances allowed when the switch is running in the 1x1 STP mode.

However, UNP dynamic VLANs differ from standard VLANs as follows:

- A dynamic VLAN cannot be deleted using standard VLAN commands. The VLAN is only removed when the UNP to which the VLAN is mapped is deleted.
- UNP dynamic VLANs are identified as a separate type of VLAN. The **vlan show** commands will display this type with the default name of "UNP-DYN-VLAN" and the designated type as "UNP Dynamic Vlan".
- Dynamic VLANs are not saved in the "! VLAN:" section of the switch configuration file (**boot.cfg**). However, the **unp** commands to enable dynamic VLAN configuration and create the UNP are saved in the "! DA-UNP:" section of the **boot.cfg** file. As a result, the VLAN is created again on the next switch bootup.

For more information, see "Enabling Dynamic VLAN Configuration" on page 30-63.

UNP Profile Attributes

In addition to profile mapping, there are configurable UNP profile attributes that are applied to device traffic once the device is moved into the profile. These attributes determine the following:

- If a list of QoS policy rules is applied to the traffic.
- If a location or time period policy restricts access to a specific location or during a specific date and time.
- Whether device traffic is redirected for internal Captive Portal authentication (the OmniSwitch serves up the login page to the user).
- Whether devices that did not pass authentication are allowed into the profile.
- Whether the UNP port to which a device is connected is tagged with the VLAN mapped to the profile when the first device is classified into that profile.
- Whether profile devices are redirected to a Unified Policy Access Manager (UPAM) server or a ClearPass Policy Manager (CPPM) server for Bring Your Own Devices (BYOD) authentication and classification.
- The bandwidth parameter values that are used to rate limit traffic on profile ports.
- The amount of time an authenticated user device remains logged into the network after the source MAC address for the device has aged out.

For more information about configuring a UNP, see "Configuring UNP Profiles" on page 30-58.

Dynamic VLAN Profiles

UNP functionality provides the ability to dynamically create VLAN profiles based on very specific traffic conditions. A UNP profile is dynamically created when the trust VLAN tag option is enabled on the UNP port or link aggregate and one of the following conditions occurs:

- A tagged packet received on the UNP port contains a VLAN tag that matches an existing MVRP VLAN in the switch configuration that is not assigned to a profile.
- There is no matching VLAN in the switch configuration.

Dynamic profiles are saved in the switch configuration, and profile attributes are configurable in the same manner as manually created profiles.

Dynamic SAP Configuration

When device traffic is assigned to a service profile, UNP first checks the switch configuration to see if a Service Access Point (SAP) already exists for the VLAN tag and other service profile attribute values that are specific to the type of service profile (SPB or VXLAN). If a SAP already exists with these values, the device traffic is classified into that SAP. If a SAP does not exist, the switch dynamically creates one based on the following SPB or VXLAN service profile attributes:

- **A VLAN tag**—This value determines the encapsulation value for the SAP (when set to zero, the VLAN ID tag of the traffic is used).
- **An SPB Service Instance ID (I-SID)**—This value is specified when configuring an SPB service mapping for a UNP profile. An I-SID is associated with an SPB service ID that is assigned to a UNP access port to form a SAP. If the I-SID value specified does not exist, the switch will dynamically create the I-SID and associated SPB service ID. After that, the SAP is dynamically configured using the dynamically created service ID.
- **An SPB Backbone VLAN (BVLAN) ID**—This value is specified when configuring an SPB service mapping for a UNP profile. A BVLAN serves as a transport VLAN for an SPB service instance associated with the SAP. If the BVLAN ID specified does not exist, the dynamic SAP is not created.
- **A VXLAN Network ID (VNID)**—This value is specified when configuring a VXLAN service mapping for a UNP profile. A VNID is associated with a VXLAN service ID that is assigned to a UNP access port to form a SAP. If the VNID value specified does not exist, the switch will dynamically create the VNID and associated VXLAN service ID. After that, the SAP is dynamically configured using the dynamically created service ID.
- **A multicast group address and/or a far-end IP address list**—These values are specified when configuring a VXLAN service mapping for a UNP profile. It is possible to configure one or both of these values for the same service mapping.
  - A multicast group address identifies the IP address of the multicast group in which the VXLAN service will participate.
  - A far-end IP address list contains a list of IP addresses that are used to dynamically create service distribution points (SDPs) for the VXLAN service. Each address represents a VXLAN tunnel endpoint (VTEP).

Allowing incoming traffic to trigger the switch to dynamically create a SAP reduces the amount of manual configuration required. This capability is similar to configuring UNP to dynamically create a VLAN based on the 802.1Q-tag of the device traffic.

Notes.

- Dynamically creating services and related SAPs is subject to available switch resources. If an attempt to dynamically create a service or SAP fails for any reason, the MAC addresses classified for the service profile are learned as filtering.
- Dynamically created SAPs are not saved to the switch configuration file.

System Default Profiles

To further automate SAP configuration, UNP also supports dynamically creating a "System Default" service profile for traffic received on UNP access ports that is *not* classified into a user-defined UNP service profile. A System Default profile specifies the attributes used to dynamically create an SPB SAP or a VXLAN SAP for the traffic.

The type of SAP that the switch will dynamically create for the System Default profile is based on the dynamic service setting for the UNP access port on which the SAP is created. For example:

- If the dynamic service port parameter is set to SPB, then a calculated SPB BVLAN, a calculated default I-SID number, and an incremental reserved service ID number are used to dynamically create a SAP for SPB service traffic received on the UNP access port.
- If the dynamic service port parameter is set to VXLAN, then a calculated VNI number, a default multicast group IP address, and an incremental reserved service ID number are used to dynamically create a SAP for VXLAN service traffic received on the UNP access port.

For information about how to configure the dynamic service port parameter, see "Configuring UNP Port Parameters" on page 30-42.

The attributes specified through the System Default profile are derived using the system parameter values and calculations described in the following sections.

SPB System Default Profile

UNP derives the System Default profile attributes as follows to dynamically create a SAP for SBP traffic:

- **I-SID number—**The I-SID number for the dynamic SAP is calculated using the following values:
  - System default I-SID number = 10,000,000
  - System default modulo number = 512
  - The VLAN tag of the traffic stream received on the UNP port (0 is used for untagged traffic).
  - The domain ID assigned to the UNP port (0 by default) multiplied by 10,000.

  Based on the above values, if traffic tagged with VLAN 30 is received on a UNP port belonging to domain 10, then the following calculation is used to determine the I-SID number:

  ```
  10,000,000 + (10 * 10,000) + (30 % 512) = 10,100,030
  ```

  The I-SID number (10,000,000) and modulo number (512) values are both user configurable. For more information about how to change these values, see "Configuring System Default Profile Parameters" on page 30-74.

- **BVLAN**—The BVLAN ID number for the dynamic SAP is calculated using the following values:
  - The I-SID value as determined by the I-SID number calculation.
  - System default BVLAN mod = 8.
  - BVLAN index = I-SID value % 8.
  - The number of existing BVLANs configured on the switch.

  Based on the above values, if there are 8 BVLANs configured on the switch (4001, 4002, 4003, 4004, 4005, 4006, 4007, 4008) and the calculated I-SID is 10100030, the calculation to determine the BVLAN ID number is as follows:

  ```
  10100030 % 8 = 6
  8[6] = 4007
  ```

In the above example, the BVLAN index is calculated as 6. This index number is then used to pick one of the 8 existing BVLANs to use for the dynamic SAP, which calculates out to 4007. As a result, BVLAN 4007 will serve as the BVLAN ID for the dynamic SAP.

- **SPB Service ID number**— SPB services are dynamically created for SPB System Default profiles. A service ID number represents the association between the calculated I-SID value and control BVLAN. A reserved service ID number (32768) is assigned to a dynamic service; this number is incremented by 1 for each additional dynamic service (SPB or VXLAN) that is created and only has local significance.
  - **Multicast Mode**—The multicast mode for a dynamic service is set to head-end.
  - **VLAN translation**—The VLAN translation status for a dynamic service is enabled.

  The SPB dynamic service parameters (multicast mode and VLAN translation) are user configurable. For more information about how to change these parameter values for dynamic services, see "Configuring System Default Profile Parameters" on page 30-74.

VXLAN System Default Profile

UNP derives the System Default profile attributes as follows to dynamically create a SAP for VXLAN traffic:

- **VNID** —The VNID number for the dynamic SAP is calculated using the following values:
  - System default VNID number = 10,000,000
  - System default modulo number = 512
  - The VLAN tag of the traffic stream received on the UNP port (0 is used for untagged traffic).
  - The domain ID assigned to the UNP port (0 by default) multiplied by 10,000.

  Based on the above values, if traffic tagged with VLAN 30 is received on a UNP port belonging to domain 10, then the following calculation is used to determine the VNID number:

  ```
  10,000,000 + (10 * 10,000) + (30 % 512) = 10,100,030
  ```

- **VXLAN Service ID number**— VXLAN services are dynamically created for VXLAN System Default profiles. A service ID number represents the calculated VNID value. A reserved service ID number (32768) is assigned to a dynamic service; this number is incremented by 1 for each additional dynamic service (SPB or VXLAN) that is created and only has local significance.
  - **Multicast Mode**—The multicast mode for a dynamic service is set to head-end.
  - **VLAN translation**—The VLAN translation status for a dynamic service is enabled.
  - **Multicast Group Address**—The multicast group IP address is set to 239.0.0.0 for a dynamic service. This address is used to build a VXLAN Service Distribution Point (SDP) tunnel for the dynamic service traffic. VXLAN nodes that subscribe to the same multicast group will receive traffic through the associated SDP tunnel from all the other VXLAN nodes that belong to the same multicast group.
  - **Far-end IP List Name**—The name of a far-end IP list to associate with a dynamic service. This list contains IP addresses each of which is assigned to the Loopback0 interface of a far-end VXLAN node. The IP addresses are used to build VXLAN SDP tunnels between the VXLAN nodes. Traffic associated with the dynamic VXLAN service is encapsulated and sent through an SDP tunnel to the destined far-end node.

  All of the dynamic VXLAN service parameters (multicast mode, VLAN translation, multicast group address, and far-end IP list name) are user configurable. For more information about how to change these parameter values for dynamic services, see "Configuring System Default Profile Parameters" on page 30-74.

System Default Profile Names

System Default profiles are automatically assigned a name when the profile is dynamically created.

- The name assigned to an SPB System Default profile is “SystemDefaultISID”, where ISID is the calculated attribute value for the profile. For example, if the calculated I-SID number is “10,100,030”, then the SPB profile “SystemDefault10100030” is created.
- The name assigned to a VXLAN System Default profile is “SystemDefaultVNID”, where VNID is the calculated attribute value for the profile. For example, if the calculated VNID number is “10,001,000”, then the VXLAN profile “SystemDefault10001000” is created.

UNP **show** commands that display profile names, will also include System Default profile names.

WLAN Access Role Profile

The defaultWLANProfile is a built-in profile that is designated for classifying OmniAccess Stellar Access Point (AP) devices. This profile is automatically assigned to a built-in LLDP MED Endpoint classification rule that will recognize active AP devices connected to the switch and will assign them to the defaultWLANProfile. The VLAN that is mapped to this profile will serve as the management VLAN for the AP devices.

Using the defaultWLANProfile to classify AP devices ensures that all of the AP devices connected to each switch in the wired network will use the same management VLAN. This profile is similar to the standard UNP VLAN profile but with the following considerations and guidelines:

- The profile cannot be deleted; it is a built-in profile that is always available in the switch configuration.
- Only the following profile attributes are configurable:
  - VLAN mapping.
  - QoS policy list assignment.
  - Authentication flag status.
  - Mobile tag status.
- The defaultWLANProfile does not appear in the configuration snapshot for the switch. However, when the default value of any of the above configurable attributes is modified, then the profile will appear in the configuration snapshot.

See “OmniAccess Stellar AP Integration” on page 30-101 for more information.

UNP Ports

Access Guardian functionality is supported only on UNP-enabled ports or link aggregates. Traffic from a device connected to a UNP port triggers an authentication and classification process that is used to determine the UNP profile assignment for the device.

By default, all switch ports are non-UNP (fixed) ports that are statically assigned to a specific VLAN. Once UNP is enabled on a port, traffic from each device connected to that port is classified using the UNP port and profile configuration to determine the VLAN or service assignment for the device.

There are two types of UNP ports: bridge and access. The port type is specified when UNP functionality is enabled on the port.

- If a port is configured as a UNP bridge port, then traffic received on that port is only classified using VLAN profiles.

- If a port is configured as a UNP access port, then traffic received on that port is only classified using service profiles.

The port type basically determines if device traffic received on that port is classified into the VLAN domain or the service domain.

When a UNP bridge port is dynamically assigned to a VLAN, a VLAN port association (VPA) is created and tracked by VLAN management software on each switch. Because the UNP configuration is applied to each device connected or forwarded through a UNP port, the UNP port can associate with more than one VLAN.

UNP access ports are not dynamically assigned to VLANs. Instead, traffic received on the port is classified to a Service Access Point (SAP). A SAP is a virtual port that maps classified device traffic to a service.

UNP Port Attributes

In addition to the UNP port type, there are configurable UNP port-level attributes that determine the following for devices connected to a UNP port or link aggregate:

- The type of device authentication (802.1X and/or MAC) attempted, if any.
- Whether device classification is enabled to move devices into profiles based on the outcome of the device authentication process. For example, authentication is not enabled or fails to determine the profile assignment for the device.
- Whether devices that do not receive a UNP profile assignment through the authentication or classification process are assigned to a default profile associated with the UNP port.
- If device traffic is segregated into logical groups based on the domain ID assigned to the UNP port.
- If a port bounce is performed on a UNP bridge port that interacts with the Unified Policy Access Manager (UPAM) or the ClearPass Policy Manager (CPPM) as part of the OmniSwitch Bring Your Own Devices (BYOD) solution.

UNP port-level attributes are different from UNP profile-level attributes as follows:

- Port-level attributes define the UNP functionality that is applied to device traffic to help determine the UNP profile assignment for the device.
- Profile-level attributes define the UNP functionality that is applied once a device is assigned to a profile. Profile attributes determine the level of access to network resources for devices assigned to the profile and whether devices are redirected to a UPAM server or a CPPM server for authentication and classification.

For more information about UNP port attributes, see "Configuring UNP Port-Based Functionality" on page 30-41.

UNP Port Domains

A UNP port domain is a configurable port-level attribute that provides an additional method for segregating device traffic. A domain is identified by a numerical ID, which can be assigned to UNP ports and profile classification rules. By default, all UNP ports (bridge and access) and profile rules are assigned to domain 0.

The main benefit of UNP port domains is that they provide the ability to group physical UNP ports or link aggregates into one logical domain. Once a UNP port is assigned to a specific domain ID, only classification rules associated with the same domain ID are applied to that port.

An example of using port domains would be to group all UNP ports carrying traffic for a specific customer into the same domain (all Customer A ports assigned to domain 2). Then, assign UNP classification rules associated with VLAN and/or service profiles tailored for that customer to the same domain ID (all profile classification rules for Customer A are assigned to domain 2).

For more information about UNP port domains, see "Configuring UNP Port Domains" on page 30-52 and "Configuring the Domain Classification Rule" on page 30-81.

UNP Classification Rules

Classifying devices with UNP rules allows the administrator to assign users to a profile group based on port and device attributes, such as source IP address, source MAC address, port, or domain ID. For example:

- Classification is enabled on UNP port 1/1/10.
- A MAC address range classification rule is associated with a UNP profile named "Engineering". This rule defines a MAC address range of "00:11:22:33:44:55 through 00:11:22:33:44:66".
- A device connecting to port 1/1/10 with a source MAC address that falls within the specified MAC address range is dynamically assigned to the "Engineering" profile. The device and the port on which the device was learned are also dynamically assigned to the VLAN or service that is associated with the profile.

Enabling classification and defining classification rules is optional with UNP. When enabled, however, classification rules are only applied to UNP port traffic when one of the following occurs:

- 802.1X and MAC authentication are disabled on the port.
- 802.1X and/or MAC authentication is enabled but the RADIUS server is not configured.
- 802.1X and/or MAC authentication is enabled but the RADIUS authentication process did not return a UNP name or failed.

If classification is disabled on a UNP port, classification rules are not applied to traffic received on that port. If both authentication and classification are disabled on a UNP port, traffic received on that port is blocked, unless a default UNP is configured for that port.

UNP Rule Types

A classification rule specifies the criteria that a device must match and the name of a UNP profile that is applied to the device when the match occurs. The following table lists all the UNP classification rules in the order of precedence (highest to lowest).

Precedence Step/Rule	Matching Condition
1. Port + VLAN tag	Packet is learned on a matching port or link aggregate *and* the packet contains a matching VLAN ID tag.
2. Port	Packet is learned on a matching port or link aggregate.
3. Domain ID + VLAN tag	Packet is learned on a port or link aggregate that is assigned to a matching domain ID *and* the packet contains a matching VLAN ID tag.
4. Domain ID	Packet is learned on a port or link aggregate that is assigned to a matching domain ID.

Precedence Step/Rule	Matching Condition
5. MAC address + VLAN tag	Packet contains a matching source MAC address *and* a matching VLAN ID tag.
6. MAC address	Packet contains a matching source MAC address.
7. MAC OUI + VLAN tag	Packet contains a source MAC address with a matching OUI and a matching VLAN ID tag.
8. MAC OUI	Packet contains a source MAC address with a matching OUI.
9. MAC address range + VLAN tag	Packet contains a source MAC address that falls within a specified range of MAC addresses *and* a matching VLAN ID tag.
10. MAC address range	Packet contains a source MAC address that falls within a specified range of MAC addresses.
11. LLDP for media endpoint devices	LLDP TLVs from an IP phone or from an OmniAccess Stellar Access Point (AP) are detected.
12. Authentication Type + VLAN tag	Packet received from a device authenticated through the matching authentication type *and* the packet contains a matching VLAN ID tag.
13. Authentication Type	Packet received from a device authenticated through the matching authentication type.
14. IP address + VLAN tag	Packet contains a matching source IP address *and* a matching VLAN ID tag.
15. IP address	Packet contains a matching source IP address.
16. VLAN tag	Packet contains a matching VLAN ID tag.

Binding Classification Rules for UNP Profiles

The port and domain ID classification rules can be combined with other classification rules to create a binding rule. The following binding rule combinations are supported and are listed in the order of precedence:

1. Port + MAC address + IP address
2. Port + MAC address
3. Port + IP address
4. Domain ID + MAC address + IP address

A device must match all the rules specified in the binding rule combination. For example, if a binding rule specifies a port, MAC address, and IP address, then the device must have a matching port, source MAC address, *and* source IP address.

Note. Binding classification rules take precedence over individual classification rules.

Extended Classification Rules for UNP Profiles

An Extended classification rule defines a list of individual rules and assigns the list a name and a precedence value. A device must match all of the rules specified in the extended rule list.

The precedence value assigned to the extended rule's name is used to determine precedence among other extended classification rules configured on the switch. If a device matches all the criteria in two different extended rules, the rule with the highest precedence is applied to the device.

Although some individual classification rules can be combined to for m a binding rule, a binding rule is not assigned a rule name and does not have a configurable precedence value. In addition, extended classification rules offer more rule combinations than binding rules.

Note. Extended classification rules take precedence over all other UNP classification rule types (individual rules and binding rules).

For more information see "Configuring UNP Classification Rules" on page 30-80.

How it Works

There is no global switch setting to invoke the Access Guardian UNP functionality. Instead, UNP is enabled on individual switch ports and profiles are defined to determine the dynamic VLAN or service assignment for devices connected through the UNP ports. When UNP is enabled on a switch port, the following device authentication and classification process is triggered when the port receives traffic.

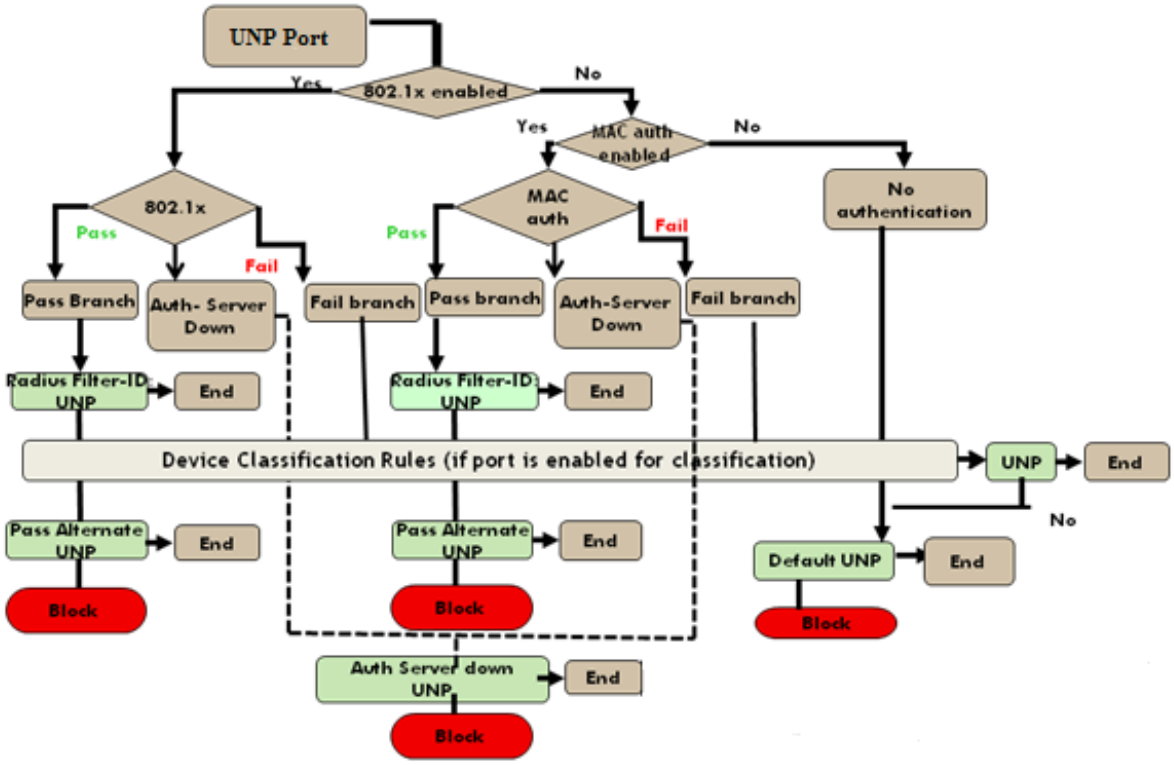


Figure 30-2 : UNP - Device authentication and classification process

Interaction With Other Features

This section contains important information about how other OmniSwitch features interact with Access Guardian. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

Authentication, Authorization, and Accounting (AAA)

The AAA configuration for the switch determines the following for Access Guardian functionality:

- Which RADIUS servers, Unified Policy Access Manager (UPAM) server, or the ClearPass Policy Manager (CPPM) server to use for Access Guardian authentication and accounting sessions.
- Authentication parameter values, such as the session timeout, inactivity timeout, interim accounting update interval, and 802.1X re-authentication interval for authentication and accounting sessions.
- AAA profiles to define a custom, pre-defined AAA configuration that can be applied to a specific set of UNP ports or through a Captive Portal profile.

Bring Your Own Devices (BYOD)

Access Guardian can interact with the Unified Policy Access Manager (UPAM) or the ClearPass Policy Manager (CPPM) to provide support for the OmniSwitch BYOD unified access solution.

- Configurable switch parameters redirect traffic to the CPPM or UPAM server.
- Configurable UNP profile parameters allow devices assigned to the profile to honor CoA and DM messages from the UPAM or CPPM.
- A port bounce operation is configurable on UNP ports to trigger re-authentication of non-supplicants upon receipt of CoA and DM messages.
- A global pause timer is available to determine the amount of time the switch filters traffic from non-supplicant (non-802.1X) devices on all UNP ports. This is done to clear the context of the user and is triggered upon receipt of a CoA message that requires a VLAN change for the device.
- Access Guardian interacts with either UPAM or CPPM for a given instance, but not both at the same time.

For more information about the OmniSwitch BYOD solution, see “Bring Your Own Devices (BYOD) Overview” on page 30-150.

Learned Port Security

UNP and Learned Port Security (LPS) are supported on the same port with the following conditions:

- LPS is not supported on link aggregates.
- The LPS learning window is not set on a per-port basis, which means that the window applies globally across all UNP ports on which LPS is enabled.
- When LPS is enabled or disabled on a UNP bridge port (LPS is not supported on UNP access ports), MAC addresses already learned on that port are flushed.

- Configuring a static MAC address is not allowed on a UNP port unless LPS is also enabled on the same port.
- When both LPS and UNP are enabled on the same port,
  - UNP first authenticates and classifies any MAC addresses received, then LPS rules are applied.
  - If a MAC address violates any of the LPS rules for the port, the address may get filtered or the port violated even if UNP initially determined the address was valid. In other words, LPS rules take precedence over UNP to determine if a MAC address is bridged or filtered on the port.
- If UNP classifies a MAC address as learning but LPS learns the address as filtering, an untagged packet will show as filtering in the default VLAN for the port and a tagged packet MAC will show as filtering in the specific tagged VLAN.
- When a MAC address is filtered by LPS, the **show unp user status** command will display "LPS-Blocked" as the classification source for that MAC address.

There are some LPS commands and command options that are not supported on UNP ports. For more information about these exceptions and other conditions for using UNP and LPS on the same port, see Chapter 47, "Learned Port Security Commands," in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Multiple VLAN Registration Protocol (MVRP)

MVRP is not supported on UNP ports, however, both features can co-exist on the same switch. The recommended configuration is to have UNP dynamically create VLAN-port-associations on UNP bridge ports while MVRP propagates the dynamic VLANs down and up stream.

UNP supports dynamic VLAN and dynamic profile configuration options that are used to facilitate the interaction between UNP and MVRP. A UNP dynamic VLAN, and optionally, a dynamic profile is created when one of the following occurs:

- A UNP profile is mapped to a VLAN ID that does not already exist in the switch configuration. A UNP dynamic VLAN is created with the specified VLAN ID.
- A UNP profile is mapped to an MVRP VLAN. The MVRP VLAN is then converted to a UNP dynamic VLAN.
- Tagged packets received on UNP ports that are enabled to trust the VLAN tag are classified based on the VLAN tag of the packet.
  - If the VLAN tag does not match any existing VLAN, a UNP dynamic VLAN matching the VLAN tag is created. At the same time, a UNP dynamic profile is created and mapped to the dynamic VLAN ID.
  - If the VLAN tag matches an MVRP VLAN on the switch and the MVRP VLAN is not already assigned to a profile, a new profile is automatically created and associated with the MVRP VLAN and the MVRP VLAN is converted to a UNP dynamic VLAN.
- If a UNP profile is mapped to an MVRP VLAN but the dynamic VLAN configuration option is disabled, the MVRP VLAN is not converted to a UNP dynamic VLAN. However, when the dynamic VLAN option is subsequently enabled, the MVRP VLAN is converted to a UNP dynamic VLAN.

Refer to "Enabling Dynamic VLAN Configuration" on page 30-63 and "Enabling Dynamic Profile Configuration" on page 30-65 for more information.

For more information about using MVRP, see Chapter 14, "Configuring MVRP," in this guide.

Quality of Service (QoS)

The Access Guardian feature provides the ability to assign a list of QoS policy rules to a UNP. The rules contained in the list are applied to any device that is assigned to the UNP. Consider the following guidelines when configuring policy lists for user profiles:

- QoS policy rules and policy lists are configured using the OmniSwitch QoS feature. Configuration of these items is required before the list is assigned to a UNP.
- Configuring QoS policy lists is not allowed if VLAN Stacking Services or if QoS inner VLAN or inner 802.1Q tag policies are configured for the switch.
- Only one QoS policy list per UNP is allowed, but multiple profiles can use the same UNP. Up to 32 policy lists (including the default list) are allowed per switch.
- A default QoS policy list always exists in the switch configuration. Any QoS policies that are not assigned to a user profile belong to the default list, unless specified otherwise when the policy is created.
- If a QoS policy list is configured for a UNP profile, only the policy rules in the list are applied to traffic from devices classified into the profile. Any default list policy rules are not applied in this case.
- If a QoS policy list is not specified for a user profile, then any policies from the default list are applied to profile devices.
- If a policy rule is enabled, it is active for all policy lists to which it belongs. If one of the policy lists is disabled, the rule is still active for all the other lists.
- If a policy rule is disabled, it is no longer active in any policy list to which it belongs, even if the list is still enabled.

Service Assurance Agent

The Service Assurance Agent (SAA) profile is particularly useful for monitoring VM connectivity across the data center. This profile type specifies jitter and latency threshold values and is assigned to UNP VLAN profiles (service profiles not supported) to associate these performance monitoring thresholds with a specific UNP.

The OmniVista network management tool will extract profile information from UNP on the switch and will create SAA sessions based on the UNP profile SAA threshold values. These SAA sessions will operate as regular sessions. When a threshold is reached, a trap is sent to OmniVista, and OmniVista will make the necessary notifications and network modifications.

Service Manager

The OmniSwitch supports both a VLAN and a service domain for traffic classification. The VLAN domain is identified by a VLAN ID. The service domain is identified by a Shortest Path Bridging (SPB) service instance identifier (I-SID) or a Virtual eXtensible LAN (VXLAN) Network ID, both of which are translated into a Service Manger service ID to represent a virtual forwarding instance (VFI).

- In the VLAN domain, each VLAN is accessed through a physical port. Each physical port can have more than one VLAN attached. UNP VLAN classification associates a MAC address to a specific VLAN on a physical UNP bridge port.
- In the service domain, each VFI is accessed through a virtual port, referred to as a Service Access Point (SAP). UNP service classification associates a MAC address to a SAP.

Source Learning

Do not disable source learning on a port or VLAN when using UNP to classify devices connected to UNP-enabled ports.

Universal Network Profile (UNP)

The UNP feature provides network administrators with the ability to define and apply network access control to specific types of devices by grouping such devices according to specific matching profile criteria. This allows network administrators to create virtual machine network profiles (vNPs) *and* user network profiles from a unified framework of operation and administration.

UNP is not limited to creating profiles for only certain types of devices. However, the following classification methods implemented through UNP functionality and profile criteria provide the ability to tailor profiles for specific devices (physical or virtual):

- MAC-based and 802.1X-based authentication using a RADIUS-capable server.
- Redirection for Captive Portal authentication.
- Redirection to the Unified Policy Access Manager (UPAM) or the ClearPass Policy Manager (CPPM) for Bring Your Own Devices (BYOD) user device registration, integrity check, UNP assignment, and policy list assignment.
- Switch-wide classification rules to classify users based on port and device attributes (for example, source MAC, Domain ID, IP address). No authentication required.
- VLAN tag classification to create VLAN port or Service Access Point (SAP) associations based on the VLAN ID contained in device packets.
- Default UNP classification for traffic not classified through other methods.

Basically, UNP functionality is used to define profile-based VLANs or services to which network devices are assigned. The profile can allow, deny, or require actions by users or machines on the network. Because membership to a VLAN or service is based on UNP profile criteria, devices assigned to the VLAN or service are not tied to a specific port or switch. This flexibility allows device mobility within the network while maintaining network security.

Virtual Network Profiles

A Virtual Network Profile (vNP) refers to a UNP that is configured for machine classification, in particular virtual machines. This type of UNP will classify virtual machines in the same manner as any other device connected to a UNP port.

Once a virtual machine is assigned to a vNP, the VM traffic is bound to the VLAN or service as defined by the profile. In addition, any QoS policies associated with the profile are also applied to the VM traffic. See “Device Authentication” on page 30-13 for more information.

For more information about virtual machine classification, see the “Virtual Machine Classification” chapter in the *OmniSwitch AOS Release 8 Data Center Switching Guide*.

UNP Port Interaction with Other Features

The following tables provides a summary list of switch features and whether or not each feature is supported on UNP-enabled ports:

Feature	UNP Port
802.1q	Not supported. Supported on untagged ports.
Application Fingerprinting (AFP) UNP mode	Supported (UNP is applied first then AFP if the UNP applies a QoS policy list rule that specifies an AFP group name).
Application Monitoring and Enforcement (AppMon)	Supported.
Ethernet OAM port	Not supported.
Ethernet Ring Protection (ERP)	Not supported.
Ethernet Services (VLAN Stacking)	Not supported.
IPv6	Not supported.
Learned Port Security (LPS)	Supported (UNP is applied first then LPS if UNP classifies the MAC address in a forwarding state).
Link Aggregation	Supported (not supported on ports that are members of a link aggregate).
Multiple VLAN Registration Protocol (MVRP)	Not supported.
OpenFlow enabled port	Not supported.
Port Mirroring	Not supported on destination ports (MTP). Supported on source ports.
Port Monitoring	Supported
Port Mapping	Not supported on network ports. Supported on user ports.
Service Manager access ports	Not supported.
Service Manager network ports	Not supported.
Source Learning	Not supported on ports on which dynamic source learning is disabled. In addition, disabling VLAN-level source learning is not recommended.
STP port enable or disable	Not supported.
Static MAC addresses	Supported only when LPS is also enabled on the UNP port.

UNP Dynamic SAPs

When a device is classified into a UNP profile that is mapped to service parameters, a Service Access Point (SAP) is dynamically created based on the service parameter values.

IPMS for UNP Dynamic SPB SAPs

Shortest Path Bridging (SPB) multicast optimization applies the functionality of IGMP/MLD snooping (OmniSwitch IP Multicast Switching) to static SPB services and associated SAPs. This allows SPB backbone edge bridges to perform multicast filtering on a per-SAP, per-service basis to ensure that IP multicast traffic is not sent out SAP ports onto LANs where there are no devices requesting to receive the multicast stream. As a result, configuring IP Multicast Switching for SPB services helps to cut down on the unnecessary forwarding of IP multicast traffic.

This same functionality can also be applied to dynamic SPB services and associated SAPs that are created through the UNP framework. IGMP and MLD snooping options are configurable mapping attributes for UNP service profiles that are mapped to SPB service parameters. When a device is classified into the SPB service-mapped profile, a dynamic SPB SAP is created and the specified IGMP/MLD snooping functionality is applied to the dynamic SAP. Refer to "Mapping Service Parameters to a UNP Profile" on page 30-66 for information about configuring IGMP and MLD snooping profile attributes.

For more information about IGMP/MLD snooping, see Chapter 27, "Configuring IP Multicast Switching."

VRRP over UNP Dynamic SPB SAPs

When a dynamic UNP SAP connects two VRRP routers over an SPB backbone service, VRRP advertisements are sent through the SPB service domain to elect one router as the master and one as the slave (backup router). The slave router does not send out VRRP advertisements; only listens for advertisements from the master router. This inactivity may cause the dynamic UNP SAP on which the slave router communicates to age out. When this occurs, the slave router will no longer receive advertisements from the master router and will elect itself as the master. This results in two dual VRRP master routers operating within the same service domain.

To support VRRP communication over UNP SAP connections, the following configuration steps are required in the order given:

1 Enable MAC address mobility for a UNP profile that is mapped to SPB service parameters. This will provide support for VRRP MAC address movement that is required for the VRRP master/slave election process. Refer to "Configuring UNP Profile Attributes" on page 30-59 for more information.

2 Statically assign the UNP service profile to the UNP access port on which a VRRP router will communicate. This will create a persistent SAP that won't age out and will ensure an uninterrupted flow of VRRP advertisements to the VRRP router. Refer to "Statically Assigning Service Profiles for Silent Devices" on page 30-55 for more information.

3 Configure the VRRP routers that will communicate over the persistent SAPs. Refer to Chapter 25, "Configuring VRRP," for more information.

Configuring Port-Based Network Access Control

For port-based network access control, the switch must know which servers to use for authenticating supplicant (802.1X) and non-supplicant (non-802.1X) user devices. In addition, the Universal Network Profile (UNP) feature must be active to perform authentication and classification functions for a supplicant and non-supplicate device.

Configuring the UNP feature consists of both profile-based and port-based configuration tasks. The profile-based tasks define profile attributes that enforce network access control for devices classified into the profile. The port-based tasks enable UNP functionality on individual ports.

The following sections describe configuring Access Guardian features to provide port-based network access control:

- "Setting Authentication Parameters for the Switch" on page 30-34.
- "Configuring UNP Port-Based Functionality" on page 30-41.
- "Configuring UNP Profiles" on page 30-58.
- "Configuring the UNP Profile Mapping" on page 30-62
- "Configuring QoS Policy Lists" on page 30-76.
- "Configuring UNP Classification Rules" on page 30-80.

Setting Authentication Parameters for the Switch

Use the **aaa device-authentication** command to specify which RADIUS servers the switch will use for 802.1X, MAC, and Captive Portal authentication. The server information must already be configured on the switch through the **aaa radius-server** command. An example of setting the switch to use specific servers for 802.1X authentication:

```
-> aaa radius-server rad1 host 10.10.2.1 key rad1_secret
-> aaa radius-server rad2 host 20.20.2.1 key rad2_secret
-> aaa device-authentication 802.1x rad1 rad2
```

In this example, the **rad1** server is used for authenticating user devices connected to UNP ports on which 802.1X authentication is enabled. If **rad1** becomes unavailable, the switch then uses **rad2** for 802.1X authentication.

To set the switch to use specific servers for MAC authentication, use the **aaa device-authentication** command with the **mac** parameter. For example:

```
-> aaa device-authentication mac rad1 rad2
```

In this example, the **rad1** server is used for authenticating user devices connected to UNP ports on which MAC authentication is enabled. As in the 802.1X authentication example, if **rad1** becomes unavailable, the switch will then use **rad2** for MAC authentication.

To set the switch to use specific servers for internal Captive Portal authentication, use the **aaa device-authentication** command with the **captive-portal** parameter. For example:

```
-> aaa device-authentication captive-portal rad1 rad2
```

In this example, the **rad1** server is used for authenticating user devices connected to UNP ports that are classified into a UNP profile that has Captive Portal authentication enabled. As in the 802.1X and MAC authentication example, if **rad1** becomes unavailable, the switch will then use **rad2** for internal Captive Portal authentication.

Note. The same RADIUS servers can be used for 802.1X, MAC, and Captive Portal authentication. Using different servers for each type of authentication is allowed but not required. For more information about configuring authentication servers, see Chapter 39, "AAA Commands," in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Use the **show aaa server** command to display the RADIUS server configuration. For example:

```
-> show aaa server
Server name = rad1
  Server type          = RADIUS,
  IP Address 1         = 10.10.2.1,
  Retry number         = 3,
  Time out (sec)       = 2,
  Authentication port  = 1812,
  Accounting port      = 1813,
  VRF                  = default
Server name = rad2
  Server type          = RADIUS,
  IP Address 1         = 20.20.2.1,
  Retry number         = 3,
  Time out (sec)       = 2,
  Authentication port  = 1812,
  Accounting port      = 1813,
  VRF                  = default
```

Use the **show aaa device-authentication** command to display a list of RADIUS servers assigned to provide 802.1X, MAC, or Captive Portal authentication. For example:

```
-> show aaa device-authentication
Authentication type = mac
    Authentication Server:
      1st authentication server = rad1,
      2nd authentication server = rad2

Authentication type = 802.1x
    Authentication Server:
      1st authentication server = rad1,
      2nd authentication server = rad2

Authentication type = captive-portal
    Authentication Server:
      1st authentication server = rad1,
      2nd authentication server = rad2
```

For more information about the authentication methods for supplicant and non-supplicant devices, see "Device Authentication" on page 30-13.

Accounting Servers

Use the **aaa accounting** command to create an accounting server entry for 802.1X, MAC, and Captive Portal authentication. For example, the following commands specify accounting servers for each type of authentication:

```
-> aaa accounting mac rad1 rad2 rad3
-> aaa accounting 802.1x rad1 rad2 rad3 rad4
-> aaa accounting captive-portal rad1 rad2 rad3
```

Optionally, the Switch Logging (syslog) facility can be used for the accounting function. For example, the following commands specify syslog as the accounting server for each type of authentication:

```
-> aaa accounting 802.1x syslog 10.135.67.99 port 8000
-> aaa accounting mac syslog 10.135.67.99 port 8000
-> aaa accounting captive-portal syslog 10.135.67.99 port 8000
```

Accounting with the local syslog facility is not allowed if RADIUS server accounting is already configured. In other words, configure either RADIUS *or* syslog accounting.

Use the **show aaa accounting** command to display the accounting server configuration for a specific type of device authentication. For example:

```
 -> show aaa accounting mac
Accounting type = mac
    Accounting Server:
      1st Accnt Server = rad1,
      2nd Accnt Server = rad2
```

Configuring Authentication Session Parameters

The following table provides a list of configurable authentication session parameters, the default value for each parameter, and the authentication type (802.1X, MAC, or Captive Portal) to which the parameter applies:

Description	Command	Default	Authentication Type
The amount of time a session remains active after a successful login	**aaa session-timeout**	Timer = disabled Time limit = 43200 seconds (12 hours)	MAC, Captive Portal
The amount of time an inactive user can remain logged on	**aaa inactivity-logout**	Timer = disabled Time limit = 600 seconds	MAC, Captive Portal
Accounting update interval	**aaa interim-interval**	Timer = disabled 600 seconds	802.1X, MAC, Captive Portal
Number of login attempts allowed per session	**captive-portal retry-count**	3	Captive Portal
The re-authentication time interval	**aaa 802.1x re-authentication**	Timer = disabled Time limit = 3600 seconds	802.1X
The port identifier for the NAS-Port attribute	**aaa radius nas-port-id**	User port	802.1X, MAC, Captive Portal
The system identifier for the NAS-Identifier attribute	**aaa radius nas-identifier**	System name of the switch.	802.1X, MAC, Captive Portal
The source IP address for the NAS-IP-Address attribute	**aaa radius nas-ip-address**	IP address of the interface used to send the RADIUS packet.	802.1X, MAC, Captive Portal
The MAC address format for the Calling-Station-Id and the Called-Station-ID attributes	**aaa radius mac-format**	No delimiter, uppercase	802.1X, MAC, Captive Portal

The **aaa session-timeout**, **aaa interim-interval**, and **aaa 802.1x re-authentication** include a **trust-radius** option that is disabled by default. When enabled, the value for the time is taken from the following RADIUS attribute values returned from the RADIUS server. For example:

- The Session-Timeout attribute value received in an Access-Accept messages is used for the session timeout and 802.1X re-authentication parameter values.
- The Acct-Interim-Interval attribute value received in an Access-Accept message is used for the accounting interim update interval parameter.

Use the **show aaa config** command to display the current authentication session parameters values for each type of authentication. For example:

```
-> show aaa mac config
Authentication type = mac
    Session Timeout:
      Status                    = disable,
      Interval (sec)            = 43200,
      Trust Radius              = disable
```

```
    Inactivity Timeout:
      Status                    = disable,
      Interval (sec)            = 600

    Accounting Interim:
      Interval (sec)            = 600,
      Trust Radius              = disable
```

Use the **show aaa radius config** command to display RADIUS client attribute values and the MAC address format. For example:

```
-> show aaa radius config
RADIUS client attributes:
  NAS port id         = default,
  NAS identifier      = default
  NAS IP address      = default,
    MAC format delimiter:
      Username            = none, UserNameCase = uppercase,
      Password            = none, PasswordCase = uppercase,
      calling station id  = none, ClgStaIdCase = uppercase,
      called station id   = none, CldStaIdCase = uppercase
```

For more information about the commands described in this section, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Enabling or Disabling Console Session

Console session helps in security-sensitive networks and deployments. The option manages the access to the switch configuration shell through the console port.

The feature allows the following operations:

- Enable or disable the access to the switch configuration shell through the console port.
- Allows storing the configuration in the configuration file so that even after a reboot, the access to the switch remains through console port.

Use the command **aaa session console** to enable the switch access through the console port through the CLI shell. Example:

```
-> aaa session console enable
```

Use the command **aaa session console** to disable the switch access through the console port through the CLI shell. Example:

```
-> aaa session console disable
```

However, when the CLI console shell is disabled, switch can be accessed through SSH or telnet or WebView session.

In case the console access is disabled through configuration (on both working and certified directory) and if the telnet/SSH/WebView session is also not available to the switch, contact customer support to recover the switch.

Note. Deleting configuration file will also delete the other configurations. Hence, it is recommended to create a back-up of the configuration file before deleting the configuration file.

Using AAA Configuration Profiles

An AAA profile is a configuration entity that provides flexible assignment of switch-based authentication parameters to specific UNP ports. When an AAA profile is assigned to a UNP port, the parameter values defined in the profile are applied to the sessions on that port. The profile configuration overrides the global AAA configuration for users authenticating on the assigned port.

Use an AAA profile to define and apply the following AAA configuration settings:

- The authentication server to use for 802.1X, MAC, and Captive Portal authentication.
- The accounting server to use for 802.1X, MAC, and Captive Portal authentication.
- Authentication session parameter values, such as the session timeout, inactivity timeout, interim accounting interval, or 802.1X re-authentication interval.
- RADIUS attribute values for NAS-Port, NAS-Identifier, and NAS-IP-Address attributes.
- MAC address format used when a MAC address is specified in the Calling-Station-ID and Called-Station-ID attributes.

AAA profiles can be used to apply different sets of AAA configuration parameters to different sets of ports. For example, different AAA profiles could be created to point to different RADIUS servers for each authentication method. This would allow the switch to interact with a specific server on one set of ports and interact with a different server on another set of ports.

In addition, an AAA profile can be assigned to a Captive Portal profile to define specific AAA configuration options for Captive Portal authentication. A Captive Portal profile is assigned to a UNP profile and applied when Captive Portal authentication is enabled for the profile.

Configuring AAA Profiles

Use the **aaa profile** command to create a profile name and configure parameter values for that profile. For example, the following commands configure specific AAA profile parameters; all of the other profile parameters that are not configured will apply the default profile settings:

```
-> aaa profile ap-1
-> aaa profile ap-1 device-authentication mac rad1 rad2
-> aaa profile ap-1 device-authentication 802.1x rad1 rad2
-> aaa profile ap-1 device-authentication captive-portal rad1 rad2

-> aaa profile ap-1 accounting 802.1x rad1 rad2
-> aaa profile ap-1 accounting mac rad1 rad2
-> aaa profile ap-1 accounting captive-portal syslog 10.135.67.99 port 8000

-> aaa profile ap-1 802.1x re-authentication enable trust-radius enable

-> aaa profile ap-1 mac inactivity-logout enable
-> aaa profile ap-1 captive-portal inactivity-logout enable interval 600
```

Use the **unp aaa-profile** command to assign an AAA profile to a UNP port or UNP link aggregate. For example:

```
-> unp port 1/1/5 aaa-profile ap-1
-> unp port 1/2/1-5 aaa-profile ap-1
-> unp linkagg 10 aaa-profile ap-1
-> unp linkagg 2-5 aaa-profile ap-1
```

Use the **captive-portal-profile** command to assign an AAA configuration profile to a Captive Portal Profile. For example:

```
-> captive-portal-profile cp_p1 aaa-profile ap-1
```

Use the **show aaa profile** command to display the AAA profile configuration. For example,

```
-> show aaa profile ap2

AAA profile name = ap2
Authentication type = mac
    Authentication Server:
      1st Auth Server   = rad1,
      2nd Auth Server   = rad2

    Accounting Server:
      1st Accnt Server  = rad1,
      2nd Accnt Server  = rad2

    Session Timeout:
      Status            = disable,
      Interval (sec)    = 43200,
      Trust Radius      = disable

    Inactivity Timeout:
      Status            = disable,
      Interval (sec)    = 600

    Accounting Interim:
      Interval (sec)    = 600,
      Trust Radius      = disable

Authentication type = 802.1x
    Re-Authentication Timeout:
      Status            = disable,
      Interval (sec)    = 3600,
      Trust Radius      = disable

    Accounting Interim:
      Interval (sec)    = 600,
      Trust Radius      = disable

Authentication type = captive-portal
    Session Timeout:
      Status            = disable,
      Interval (sec)    = 43200,
      Trust Radius      = disable

    Inactivity Timeout:
      Status            = disable,
      Interval (sec)    = 600

    Accounting Interim:
      Interval (sec)    = 600,
      Trust Radius      = disable

RADIUS client attributes:
  NAS port id       = default,
  NAS identifier    = default,
  NAS IP address    = default,
    MAC format delimiter:
      Username           = none, UserNameCase = uppercase,
```

```
Password           = none, PasswordCase = uppercase,
calling station id = none, ClgStaIdCase = uppercase,
called station id  = none, CldStaIdCase = uppercase
```

For more information about the commands described in this section, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring a Delayed Learning Time Interval

Configuring a delayed learning interval gives the switch time to bring up IP interfaces and time for route convergence to complete before any attempt to reach an authentication server is made. By default, there is no delay time set. To configure a delay time interval value (in seconds), use the **unp delay-learning** command.

```
-> unp delay-learning 250
```

When the delay learning time value is set, the time interval is triggered when the switch boots up. During this time, any packets received on all UNP ports are dropped until the timer expires

To disable the delay time interval, set the value to zero.

```
-> unp delay-learning 250
```

Use the **show unp global configuration** command to verify if the delayed learning time interval is set.

Configuring an Authentication Server Down UNP

An authentication server down UNP is used to classify devices attempting to authenticate through UNP ports when the RADIUS server is unreachable. By default, there is no such profile configured for the switch. To create this type of UNP, use the **unp auth-server-down** command.

```
-> unp auth-server-down profile1 down_unp
```

After a device is classified into the VLAN for this UNP, an attempt to re-authenticate the device is made after a specific period of time (60 seconds by default). To change this time value, use the **unp auth-server-down-timeout** command.

```
-> unp auth-server-down-timeout 120
```

Configuring an authentication server down UNP is highly recommended when MAC or 802.1X authentication is enabled on any UNP port or link aggregate. This is because after a switch reload, the traffic from devices connected to UNP ports and link aggregates reaches the switch and triggers the authentication process before route convergence has completed and the server can be reached.

- If an authentication server down UNP is configured, devices are temporarily learned in that profile and authentication is automatically attempted again after the timeout period expires. This allows time for the server to become reachable from the switch after a reload.
- If an authentication server down UNP is not configured, devices are learned as filtering and will remain in that state. There is no further attempt to authenticate these devices again.

The authentication down UNP and related timer value are applied to all traffic received on all UNP ports in the event the RADIUS server becomes unreachable. To verify if this setting is enabled or disabled, use the **show unp global configuration** command. For example:

```
-> show unp global configuration
  Dynamic Vlan Configuration       = Disabled,
  Dynamic Profile Configuration    = Disabled,
  Auth Server Down Profile1        = -,
  Auth Server Down Profile2        = -,
  Auth Server Down Profile3        = -,
  Auth Server Down Voice Profile1  = -,
  Auth Server Down Voice Profile2  = -,
  Auth Server Down Voice Profile3  = -,
  Auth Server Down Port Bounce     = Disabled
  Auth Server Down Timeout         = 60,
  Redirect Port Bounce             = Enabled,
  Redirect Pause Timer             = -
  Redirect http proxy-port         = 8080
  Redirect Server FQDN             = cppm.abc.com
  Redirect Server IP               = 10.135.20.50
  Allowed IP                       = -
  Force L3-Learning                = Disabled
  Force L3-Learning Port Bounce    = Enabled
  802.1x Pass Through Mode         = Disabled
  AP Mode                          = Enabled
  System-default service-mod       = 512
  System-default service-base      = 10000000
  System-default Multicast-Mode    = Headend
  System-default Vlan-Xlation      = Enabled
  System-default Multicast-Group   = 239.0.0.0
  System-default far-end-ip-list   = -
  IPv6 Drop Packets                 = Disabled,
  Delayed Learning Interval         = 0,
  Global Mac-Mobility               = Disabled,
```

Note. When device authentication fails due to an unreachable RADIUS server, an event message is sent to the switch logging utility (swlog). See Chapter 51, “Switch Logging Commands,” for more information.

Configuring UNP Port-Based Functionality

Access Guardian provides network access and QoS on a per-user basis through the framework of the Universal Network Profile (UNP) feature. UNP functionality is enabled and applied on switch ports or link aggregates. Devices connected to a UNP-enabled port or link aggregate are subject to authentication and classification as determined by the UNP port and switch configuration.

By default, UNP functionality is disabled on all switch ports and link aggregates. There are two UNP port types supported: bridge and access. To enable UNP functionality and specify a port type, use the **unp port-type** command. For example:

```
-> unp port 1/1/12 port-type bridge
-> unp linkagg 5 port-type bridge
-> unp port 1/1/13 port-type access
-> unp linkagg 6 port-type access
```

To remove the UNP configuration from a port or link aggregate, use the **no unp port** or **no unp linkagg** command. For example:

```
-> no unp port 1/1/3
-> no unp linkagg 10
```

To change the port type of an existing UNP port, remove the current UNP configuration using the **no unp port** or **no unp linkagg** command then use the **unp port-type** command to set the new port type. For example:

```
-> no unp port 1/12
-> unp port 1/12 port-type access
-> no unp linkagg 5
-> unp linkagg 5 port-type access
```

Configuring UNP Port Parameters

The UNP port parameter values listed in "Access Guardian UNP Port Defaults" on page 30-5 are applied when UNP functionality is enabled on a port or link aggregate. To change the default UNP port parameter values, use the commands listed in the following table:

Command	Description
unp redirect port-bounce	Configures the redirect port bounce status for the port. When enabled, a port bounce is triggered upon receipt of Change of Authorization (CoA) or Disconnect request (DM) messages. This command applies only to UNP bridge ports.
unp 802.1x-authentication	Configures the status of 802.1X authentication for the UNP port.
unp 802.1x-authentication pass-alternate	Assigns the name of an existing UNP as an alternate profile. If successful 802.1X authentication does not return a UNP, the device can be classified into this alternate profile.
unp 802.1x-authentication bypass-8021x	Configures whether to bypass 802.1X authentication on the port. See "Configuring 802.1X Authentication Bypass" on page 30-48.
unp 802.1x-authentication failure-policy	Configures whether to attempt MAC authentication if 802.1X authentication fails or let the port configuration classify the device.
unp 802.1x-authentication tx-period	Configures the re-transmission time interval for UNP ports on which 802.1X authentication is enabled.
unp 802.1x-authentication supp-timeout	Configures the amount of time the switch will wait before timing out an 802.1X user attempting to authenticate through the port.
unp 802.1x-authentication max-req	Configures the maximum number of times the switch will transmit a request for authentication information to an 802.1X user on the port.
unp mac-authentication allow-eap	Configures whether to attempt 802.1X authentication after MAC authentication passes or fails on a UNP port that has 802.1X bypass enabled.
unp mac-authentication	Configures the status of MAC authentication for the UNP port.
unp mac-authentication pass-alternate	Assigns the name of an existing UNP as an alternate profile. If successful MAC authentication does not return a UNP, the device can be classified into this alternate profile.
unp classification	Configures the status of rule-based classification for the UNP port. When enabled, UNP classification rules are applied if device authentication does not provide a UNP name for a device connected to the port.

unp trust-tag	Configures the option of whether to trust the VLAN ID of a tagged packet to determine how the packet is classified. When enabled, packets carrying a VLAN ID tag that matches a VLAN configured on the switch are dynamically assigned to that VLAN.
unp default-profile	Assigns the name of an existing UNP as the default profile for the UNP port. If device authentication or classification does not provide a UNP name for a user device, the device can be classified into the default profile.
unp domain	Assigns a UNP port to a numerical domain ID. All UNP ports assigned to the same domain ID are considered members of a logical domain group. See "Configuring UNP Port Domains" on page 30-52.
unp aaa-profile	Assigns the name of an existing AAA configuration profile to a UNP port. The port-level AAA profile configuration overrides the global AAA configuration for the switch. See "Using AAA Configuration Profiles" on page 30-38.
unp port port-template	Assigns the name of a custom port template to a UNP port. By default, the "bridgeDefaultPortTemplate" template is assigned to UNP bridge ports and the "accessDefaultPortTemplate" template is assigned to UNP access ports. Use this command to assign a custom port template that will override the default port template values. See "Using UNP Port Templates" on page 30-45.
unp direction	Configures whether egress broadcast, unknown unicast, or multicast traffic is allowed on the UNP port.
unp admin-state	Configures the administrative status of the UNP configuration for the port. By default, the status is enabled. When disabled, the UNP configuration is retained but not active for port traffic.
unp dynamic-service	Configures whether the System Default service profile dynamically creates an SPB Service Access Point (SAP) or a VXLAN SAP based on the traffic received on the UNP access port. This command applies only to UNP access ports. See "System Default Profiles" on page 30-20.
unp l2-profile	Assigns the name of an existing Layer 2 profile to a UNP access port. This profile determines how Layer 2 protocol frames received on the access port are processed. By default, the Layer 2 profile "unp-def-access-profile" is assigned when a port is configured as a UNP access port. See "Configuring Layer 2 Profiles for UNP Access Ports" on page 30-53.
unp vlan	Configures an untagged or tagged VLAN-port association between the specified UNP bridge port and VLAN ID. Assigning a static VLAN is particularly useful for silent devices that are connected to the a UNP bridge port. See "Configuring UNP for Silent Devices" on page 30-55.
unp port profile	Assigns an existing service profile as a static profile for a UNP port. This type of profile assignment is particularly useful for silent devices that are connected to a UNP access port; the profile SAP won't age out when the device goes idle. See "Configuring UNP for Silent Devices" on page 30-55.

unp port ap-mode	Configures the Access Point (AP) mode status for a UNP bridge port (not supported on UNP access ports). See "OmniAccess Stellar AP Integration" on page 30-101.

Consider the following guidelines when configuring UNP port parameters:

- Any configuration change to a UNP-enabled port will flush all MAC addresses learned on that port. This applies only to CLI commands used to configure UNP port parameters.
- The UNP name specified with the **unp default-profile**, **unp 802.1x-authentication pass-alternate**, **unp mac-authentication pass-alternate**, and **unp port profile** commands must already exist in the switch configuration. See "UNP Profiles" on page 30-16 for more information.
- The default UNP for a port is basically a "last resort" UNP for traffic that was not successfully classified through other methods. If all other methods fail and a default UNP is not configured for the port, device traffic is blocked on that port.
- Parameter values defined in a custom UNP port template override the existing UNP port configuration. Any attempt to explicitly configure a UNP port parameter for a port that is associated with a custom template is not allowed. See "Using UNP Port Templates" on page 30-45 for more information.
- Enabling both 802.1X and MAC authentication is allowed on the same port, but 802.1X authentication is attempted first unless 802.1X authentication bypass is also enabled for the port. See "Configuring 802.1X Authentication Bypass" on page 30-48 for more information.
- There are two methods for configuring and applying port bandwidth parameter values to UNP ports that are assigned to a profile: QoS policy list rules and UNP profile bandwidth parameters. See "Configuring UNP Port Bandwidth" on page 30-50 for more information.
- If there is no authentication type enabled for the UNP port, then the source MAC address of a device connected to the port is not sent to the designated RADIUS server for identification and authentication. Instead, other classification parameters configured for the port are applied to the device.

Verifying the UNP Port Configuration

Use the **show unp port config** command to display the UNP port configuration. For example:

```
-> show unp port 1/1/10 config
Port  1/1/10
  Port-Type                      = BRIDGE,
  Redirect Port Bounce           = Disabled,
  802.1x authentication          = Enabled,
  802.1x Pass Alternate Profile  = -,
  802.1x Bypass                  = Disabled,
  802.1x failure-policy          = default,
  Mac-auth allow-eap             = -,
  Mac authentication             = Enabled,
  Mac Pass Alternate Profile     = -,
  Classification                 = Enabled,
  Trust-tag                      = Enabled,
  Default Profile                = -,
  Port Domain Num                = 0,
  AAA Profile                    = -,
  Port Template                  = bridgeDefaultPortTemplate,
  Port Control Direction         = Both,
  Egress Flooding                = Not Allowed,
  Admin State                    = Enabled,
  Dynamic Service                = -,
```

```
PVLAN Port Type                 = -,
Force L3-Learning               = Disabled,
Force L3-Learning Port Bounce   = Enabled,
802.1x Parameters:
        Tx-Period         = 30,
        Supp-Timeout      = 30,
        Max-req           = 2
L2 Profile                      = -,
```

Use the **show unp port configured-vlans** command to display the VLANs assigned to UNP bridge ports or link aggregates. For example:

```
-> show unp port configured-vlans
Port     Vlan   Type
-------+------+---------
0/10      500  unpUntag
0/10      501  unpUntag
1/1/10    600  unpQtag
1/1/11    601  unpUntag
1/1/11    602  unpQtag
1/1/11    603  unpQtag
```

Use the **show unp port profile** command to display the service profiles that are statically assigned to UNP ports or link aggregates. For example:

```
-> show unp port profile
Port    Profile
-------+----------------
1/1/5   static-spb1
1/1/5   static-spb2
1/1/10  static-vxlan1
1/1/10  static-vxlan2
1/1/20  static-l2gre
```

For more information about the commands described in this section, see the "Access Guardian Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Using UNP Port Templates

A UNP port template is a configuration entity that provides flexible assignment of a pre-defined UNP port configuration to specific ports. Using a port template to configure UNP functionality on a port or link aggregate avoids having to configure each parameter with a separate CLI command. Applying a port template also provides an easy way to replicate a specific configuration on multiple UNP ports.

A UNP port template is used to define and apply the UNP port configuration settings that are described in "Configuring UNP Port Parameters" on page 30-42.

Note. When a custom port template is assigned to a UNP port, the parameter values defined in the template will override any existing UNP port configuration. In addition, any attempt to explicitly configure a UNP port parameter for a port that is associated with a custom template is not allowed.

Default Port Templates

There are two default UNP port templates: "bridgeDefaultPortTemplate" (applied to UNP bridge ports) and "accessDefaultPortTemplate" (applied to UNP access ports). These templates define a default set of port parameter values that are applied at the time a port or link aggregate is configured as a UNP bridge or access port. The default templates cannot be deleted, but the template parameter values are configurable.

Configuring Port Templates

Configuring a custom UNP port template is supported. This is particularly useful when different parameter values are required for one or more UNP ports. To create a custom port template, use the **unp port-template** command. For example, the following commands create two custom port templates ("portTemplate-1" and "portTemplate-2") and configure parameter values for each template:

```
-> unp port-template portTemplate-1
-> unp port-template portTemplate-1 mac-authentication
-> unp port-template portTemplate-1 mac-authentication pass-alternate AltUNP
-> unp port-template portTemplate-1 classification
-> no unp port-template portTemplate-1

-> unp port-template portTemplate-2 802.1x-authentication
-> unp port-template portTemplate-2 classification
-> unp port-template domain 10
-> no unp port-template portTemplate-2
```

Use the **show unp port-template** command to display the UNP port template configuration. For example:

```
-> show unp port-template portTemplate-1 config
Port Template: portTemplate-1
  802.1x Authentication           = Disabled,
  802.1x Pass Alternate Profile   = -,
  Mac Authentication              = Enabled,
  Mac-Auth Pass Alternate Profile = AltUNP,
  Classification                  = Enabled,
  Trust-tag                       = Enabled,
  Default Profile                 = -,
  Port Domain Number              = 0,
  AAA-Profile                     = ,
  Redirect Port Bounce            = Disabled,
  Port Control Direction          = Both,
  802.1x Tx-Period                = 0,
  802.1x Supp-Timeout             = 0,
  802.1x Max-Req                  = 2,
  802.1x Bypass                   = Disabled,
  802.1x failure-policy           = default,
  Mac-auth allow-eap              = -,
  Force L3-Learning               = Disabled
  Force L3-Learning Port Bounce   = Disabled
  Admin State                     = Enabled,
  Dynamic Service                 = -,
  L2 Profile                      = -,
   AP Mode                         = Enabled,
```

Use the **unp port port-template** command to assign a port template to a UNP port or UNP link aggregate. For example:

```
-> unp port 1/1/5 port-template portTemplate-1
-> unp port 1/2/1-5 port-template portTemplate-2
-> unp linkagg 10 port-template portTemplate-1
-> unp linkagg 10-50 port-template portTemplate-2
```

Use the **no** form of the **unp port port-template** command to remove a template from the port. For example:

```
-> no unp port 1/1/5 port-template
-> no unp linkagg 10 port-template
```

Consider the following when removing a port template from a UNP port or link aggregate:

- When a custom template is removed (for example, the “portTemplate-2”) from a UNP port, the port reverts back to using the default template (“bridgeDefaultPortTemplate” for bridge ports or “accessDefaultPortTemplate” for access ports) to define UNP port parameter options.
- When a default template is removed (for example, “bridgeDefaultPortTemplate”) from a UNP port, the UNP port parameter options for that port are individually defined through explicit commands. For example, the following commands change the MAC authentication and classification parameters for UNP port 1/1/5 (there is no template assigned to port 1/1/5):

```
-> unp port 1/1/5 mac-authentication
-> unp port 1/1/5 classification
```

To see the name of the template that is assigned to a port, use the **show unp port config** command. For example:

```
-> show unp port 1/1/5 config
Port  1/1/5
  Port-Type                       = BRIDGE,
  Redirect Port Bounce            = Disabled,
  802.1x authentication           = Enabled,
  802.1x Pass Alternate Profile   = -,
  802.1x Bypass                   = Disabled,
  802.1x failure-policy           = default,
  Mac-auth allow-eap              = -,
  Mac authentication              = Enabled,
  Mac Pass Alternate Profile      = -,
  Classification                  = Enabled,
  Trust-tag                       = Enabled,
  Default Profile                 = -,
  Port Domain Num                 = 0,
  AAA Profile                     = -,
  Port Template                   = portTemplate-1,
  Port Control Direction          = Both,
  Egress Flooding                 = Not Allowed,
  Admin State                     = Enabled,
  Dynamic Service                 = -,
  PVLAN Port Type                 = -,
  802.1x Parameters:
          Tx-Period         = 30,
          Supp-Timeout      = 30,
          Max-req           = 2
```

If there is no template assigned to a UNP port, the “Port Template” field is blank. For example:

```
-> show unp port 1/1/11 config
Port  1/1/11
  Port-Type                       = Access,
  802.1x authentication           = Enabled,
  802.1x Pass Alternate Profile   = -,
  802.1x Bypass                   = Disabled,
  802.1x failure-policy           = default,
  Mac-auth allow-eap              = -,
  Mac authentication              = Enabled,
  Mac Pass Alternate Profile      = -,
  Classification                  = Enabled,
  Trust-tag                       = Enabled,
  Default Profile                 = -,
```

```
Port Domain Num                = 0,
AAA Profile                    = -,
Port Template                  = -,
Admin State                    = Enabled,
Dynamic Service                = spb,
PVLAN Port Type                = -,
802.1x Parameters:
        Tx-Period         = 30,
        Supp-Timeout      = 30,
        Max-req           = 2
```

Modifying Port Templates

Modifying UNP port parameter values that are applied through an existing port template is allowed. Consider the following guidelines when changing template parameter values:

- Changing any template parameter value automatically applies the new value to all UNP ports to which the template is assigned. This provides a quick and efficient method for modifying port parameters across a large number of UNP ports all at once.
- Any attempt to explicitly configure a UNP port parameter for a port that is associated with a custom template is not allowed. For example, when the explicit command is given to enable classification on port 1/1/12 but a custom template is already assigned to that port, an error message is displayed:

  ```
  -> unp port 1/1/12 classification
  ERROR: Port Template already enforced on port, please remove it for manual
  config on Port
  ```

- Explicitly changing a UNP port parameter value for a port to which one of the default templates is assigned (“bridgeDefaultPortTemplate” or “accessDefaultPortTemplate”) removes the default template assignment for that port. All port parameter options for that port will then require explicit commands to change any of the parameter values, until the next time a template is assigned to that port.

For more information about the commands described in this section, see the “Access Guardian Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring 802.1X Authentication Bypass

When a device is connected to a UNP port that has both 802.1X authentication and MAC authentication enabled, the switch first attempts to identify and authenticate the device using 802.1X EAP frames. If the device does not respond to EAP frames sent by the switch after a configurable number of attempts, then the device is identified as a non-supplicant and undergoes MAC authentication.

In some cases, however, the network administrator may want to apply MAC authentication first to all devices (supplicant or non-supplicant) connected to the UNP port. In other words, the switch does not initiate 802.1X authentication; EAP frames are not sent and any EAP frames received are ignored.

The advantage to applying MAC authentication first is that the MAC address of the device is initially verified (for example, checked against a RADIUS black list). Based on the outcome of the MAC authentication, the user device is then classified accordingly or can undergo subsequent 802.1X authentication.

To enforce MAC authentication as the initial authentication method for all devices connected to a UNP port, an 802.1X bypass operation is provided. In addition, the bypass operation provides configurable options that are used to specify if subsequent 802.1X authentication is performed on the device based on the results of MAC authentication.

Configuring 802.1X authentication bypass is done using the **unp 802.1x-authentication bypass-8021x** and **unp mac-authentication allow-eap** commands. The **unp 802.1x-authentication bypass-8021x** command enables or disables the bypass operation. The following **unp mac-authentication allow-eap** command parameters determine if subsequent 802.1X authentication is attempted on the device after MAC authentication:

- **pass**—802.1X authentication is attempted if the device passes the initial MAC authentication. If the device fails MAC authentication, 802.1X authentication is bypassed (EAP frames are ignored) and the device is classified as a non-supplicant.
- **fail**—802.1X authentication is attempted if the device fails the initial MAC authentication. If the device passes MAC authentication, 802.1X authentication is bypassed (EAP frames are ignored) and the device is classified as a non-supplicant.
- **noauth**—802.1X authentication is automatically attempted if there is no MAC authentication available for the port.

Configuration Guidelines

Consider the following guidelines before configuring 802.1X authentication bypass:

- The 802.1X bypass operation is only supported on UNP ports with 802.1X authentication enabled. See "Configuring UNP Port-Based Functionality" on page 30-41 for more information about configuring the access control mode.
- If a port has supplicants connected and 802.1X bypass is enabled for that port, the supplicants are automatically logged off to undergo authentication according to the enabled bypass configuration.
- When the 802.1X bypass configuration is modified or disabled, any non-supplicant devices are automatically logged off the port. This will free up those devices to undergo the authentication specified by the new bypass configuration.
- If re-authentication is configured for the UNP port and 802.1X bypass is enabled, the MAC authentication followed by 802.1X authentication is initially performed as configured. However, only 802.1X authentication is performed during the re-authentication process, so there is no recheck to see if the MAC address of the user device is restricted.
- Enabling 802.1X bypass is not allowed on UNP ports that are configured with an 8021X failure policy.
- When successful MAC authentication returns a UNP and the 802.1X bypass operation is configured to initiate 802.1X authentication when a device passes MAC authentication, the device is *not* moved into that UNP. Instead, the device is moved into the UNP returned by 802.1X authentication. If 802.1X authentication does not provide such information, the device is moved based on the UNP port-based configuration.
- When 802.1X bypass is enabled and after MAC authentication, the port will be in a waiting state until the 802.1X authentication process complete.
- When 802.1X bypass is enabled but the allow EAP option is not configured, then subsequent 802.1X authentication is not performed. Only the initial MAC authentication is performed and the device is classified as a non-supplicant.

Configuration Example: 802.1X Bypass with MAC Authentication Fail Policy

The following CLI configuration example enables 802.1X authentication bypass on port 2/1 and triggers subsequent 802.1X authentication if the initial MAC authentication process fails:

```
-> unp port 2/1 802.1x-authentication bypass-802.1x
-> unp port 2/1 mac-authentication allow-eap fail
```

In this example, the Access Guardian authentication process for a device connected to UNP port 2/1 is as follows:

- MAC authentication is triggered when the first frame from the new user is received, whether it is an EAP frame or not.
- EAP frames for this user are ignored until MAC authentication completes (RADIUS returns an Access-Accept or a Access-Reject response).
- If the initial MAC authentication passes (Access-Accept), 802.1X authentication is bypassed for this user and all EAP frames are ignored.
- If the initial MAC authentication fails (Access-Reject), 802.1X authentication is attempted for the user. During this transition, the EAP frames are allowed and the switch must force the supplicant to restart a fresh EAP session by sending a multicast Request Identity EAPOL on the port. This is because the supplicant may have already sent an EAPOL Start.

Configuring UNP Port Bandwidth

The following two methods are available to configure and apply port bandwidth parameter values to UNP ports that are assigned to a profile:

- **QoS policy list rules.** A QoS policy list assigned to a UNP profile applies policy rules to all traffic that is classified into that profile. For example, the following commands create a QoS policy list with rules to apply rate limiting parameters to all device ports assigned to the “UNP-1” profile:

  ```
  -> policy condition ip_traffic2 source ip 10.10.5.3
  -> policy action flowShape maximum bandwidth 10m
  -> policy action burst maximum depth 1m
  -> policy rule rule2 condition traffic2 action flowShape action burst
  -> policy list rate-limit type unp
  -> policy list rate-limit rules rule2
  -> unp profile UNP-1
  -> unp profile UNP-1 qos-policy-list rate-limit
  -> unp profile UNP-1 map vlan 50
  ```

  See “Configuring QoS Policy Lists” on page 30-76 for more information.

- **Profile bandwidth parameters.** Configurable bandwidth parameter values associated with a UNP profile are applied to traffic that is classified into the profile. For example, the following commands define profile bandwidth parameters to rate limit traffic on all device ports assigned to the “UNP-1” profile.

  ```
  -> unp profile UNP-1 maximum-ingress-bandwidth 10M
  -> unp profile UNP-1 maximum-egress-bandwidth 10M
  -> unp profile unp-1 maximum-ingress-depth 1
  -> unp profile unp-1 maximum-egress-depth 1
  ```

  See “Configuring UNP Profiles” on page 30-58 for more information.

UNP Port Bandwidth Configuration Guidelines

Consider the following guidelines when configuring UNP port bandwidth:

- The maximum ingress and egress bandwidth values are configured in Kbps or Mbps.
- The maximum ingress and egress depth values are configured in Kbps.
- The default value for the maximum ingress and egress depth settings is calculated by dividing the maximum ingress or egress bandwidth value by 25. For example, if the ingress bandwidth value is set to 500K, then the ingress depth value defaults to 20K (500K/25=20K). However, if this calculation results in a value of 0 or 1, then the default value is set to 2K.
- “Per user" bandwidth profiling is not supported. If multiple user devices are classified into different profiles but learned on the same UNP port, the bandwidth parameter values obtained for the last user learned are applied on the port. Parameter values applied through previously learned users are overwritten.
- Runtime modification of UNP ingress or egress bandwidth is allowed; the modified values are then applied to both new and already authenticated user devices learned on the profile. The new runtime values become the latest bandwidth values for the profile, so they are applied to all user devices associated with the profile.
- The bandwidth limitation applied on a port through UNP classification is not removed when a user logs out or ages out. An administrator can override the bandwidth limitation through the **qos port** command or by removing the UNP configuration on the port.
- If any port bandwidth parameter value defined for a UNP profile is modified, then the other parameters need to be configured again, otherwise they will be set to their default values. For example, consider a UNP profile with maximum egress bandwidth set to 100M and egress depth set to 10K and the maximum bandwidth is changed to 200M. In this scenario, only the modified maximum bandwidth is considered, but the egress depth is reset to the default value unless the required value is specifically configured again.
- If port bandwidth values are applied through a UNP profile *and* through a QoS policy list associated with the same profile, then the minimum of these two values is applied to the UNP port. For example, consider a UNP profile with the maximum ingress bandwidth parameter set to 200M but the QoS policy list associated with the same profile sets the maximum ingress bandwidth to 100M, then the bandwidth value of 100M is applied.

Multiple User Authentication on the Same Port

If multiple users are authenticated through the same UNP port and are classified with either RADIUS returned attributes or through locally configured classification methods, then the bandwidth associated to the latest authenticated user will override the previous bandwidth settings. If there is no bandwidth associated with the new user, then no rate limitations are enforced and the previously set bandwidth is applied to the new authenticated user.

There is no priority between bandwidth limits that are applied through the **qos port** command or applied through UNP parameters. The latest change will over write the previous bandwidth limitation applied on the port. For example:

Bandwidth Profile	Action
If a user authenticates into a UNP with no UNP bandwidth profile (no bandwidth parameters or QoS policy list to apply rate limitations).	The port bandwidth setting is applied.

Bandwidth Profile	Action
If a user authenticates into a UNP with a bandwidth profile (bandwidth parameters or QoS policy list applies rate limitations).	The UNP bandwidth setting overrides the port bandwidth setting.

Note. The same bandwidth behavior applies when the user is authenticated with QoS port bandwidth: the QoS port configuration is the latest configuration.

Configuring UNP Port Domains

UNP port domains provide an additional method for segregating device traffic. A domain is identified by a numerical ID that can be assigned to UNP ports and profile classification rules. By default, all UNP ports and profile rules are assigned to domain 0.

The main benefit of UNP port domains is that they provide the ability to group physical UNP ports or link aggregates into one logical domain. Once a UNP port is assigned to a specific domain ID, only classification rules associated with the same domain ID are applied to that port.

By default, all UNP ports are assigned to domain 0. To add additional domain IDs, use the **unp domain description** command. For example, the following command creates domain 2 with an optional description:

```
-> unp domain 2 description "Customer A Domain"
```

If the optional **description** parameter is not specified, the description defaults to "UNP Domain *x*", where *x* is the domain ID number. In the above example, if the "Customer A Domain" description was not specified with the command, the description text would default to "UNP Domain 2".

To assign UNP ports to a customer domain ID, use the **unp domain** command. For example:

```
-> unp port 1/1-3 domain 2
-> unp linkagg 5 domain 2
```

Use the **show unp domain** command to display the UNP domain ID configuration. For example:

```
-> show unp domain

Domain   Description
--------+-----------------------------------
0        Default-Domain
1        UNP Domain 1
2        Customer A Domain
```

Use the **show unp port** command to display the domain ID assignment for a UNP port. For example:

```
-> show unp port
Port  Port    Type   802.1x  Mac      Class.   Default    802.1X     MAC       Trust-Tag
      Domain         Auth    Auth                         Pass-Alt   Pass-Alt
-----+------+------+--------+--------+--------+----------+----------+---------+---------
1/15       2 Bridge Disabled Disabled Enabled  unp-1001   -          -         Disabled
1/16       2 Bridge Disabled Disabled Disabled unp-1001   -          -         Disabled
1/17       2 Access Disabled Enabled  Enabled  spb1001    -          -         Enabled
1/18       2 Access Disabled Enabled  Disabled -          -          -         Disabled
1/19       2 Bridge Enabled  Enabled  Disabled DefUnp     1XProf1    MacPAS    Enabled
1/20       2 Access Enabled  Disabled Enabled  -          1XProf2    -         Enabled
```

Configuration Example

The following CLI configuration example groups ports assigned to Customer A into UNP domain 2 and creates a MAC address range classification rule that is also associated with domain 2:

```
-> unp domain 2 description "Customer A Domain"
-> unp port 1/15-20 port-type bridge
-> unp port 1/15-20 domain 2
-> unp profile CustA
-> unp classification mac-range 00:11:22:33:44:66 00:11:22:33:44:77 domain 2
profile1 CustA
```

In this example:

- UNP domain 2 is created with a description and UNP ports 1/15-20 are assigned to domain 2.
- A UNP MAC address range classification rule is defined and associated with domain 2 and the "CustA" profile.
- When traffic is received from devices connected to ports 1/15-20, the switch determines if there are any classification rules associated with domain 2 and applies that rule to the traffic. Because UNP ports 1/15-20 belong to domain 2, the MAC address range rule is applied to traffic received on those ports.
- The source MAC address of device traffic received on ports 1/15-20 is examined to see if it falls within the range of addresses defined in the MAC address range rule. If the source MAC address of the device does fall within the specified range, the device is then assigned to the "CustA" profile.
- Network access control attributes configured for the "CustA" profile are then applied to device traffic assigned to that profile.

Configuring Layer 2 Profiles for UNP Access Ports

A Layer 2 profile determines how control frames received on a UNP access port are processed. When a port is configured as a UNP access port, a default Layer 2 profile (**unp-def-access-profile**) is applied to the port with the following default values for processing control frames:

Protocol	Default
STP	tunnel
802.1x	peer
802.3ad	peer
802.1ab	drop
GVRP	tunnel
MVRP	tunnel
AMAP	drop

If the default profile values are not sufficient, use the **service l2profile** command with the **tunnel**, **drop**, and **peer** options to create a new profile. For example, the following command creates a profile named "DropL2":

```
-> service l2profile DropL2 stp drop gvrp drop 802.1ab drop
```

Consider the following when configuring Layer 2 profiles:

- Not all of the control protocols are currently supported with the **peer**, **tunnel**, and **drop** parameters. Use the following table to determine the parameter combinations that are supported:

Protocol	Reserved MAC	peer	discard	tunnel
STP	01-80-C2-00-00-00	no	yes	yes
802.1x	01-80-C2-00-00-03	yes	yes	yes
802.1ab	01-80-C2-00-00-0E	yes	yes	yes
802.3ad	01-80-C2-00-00-02	yes	no	no
GVRP	01-80-C2-00-00-21	no	yes	yes
MVRP	01-80-C2-00-00-21	no	yes	yes
AMAP	00-20-DA-00-70-04	yes	yes	no

- When a profile is created, the new profile inherits the default profile settings for processing control frames. The default settings are applied with the new profile unless they are explicitly changed. For example, the profile "DropL2" was configured to discard STP, GVRP, and 802.1ab frames. No other protocol settings were changed, so the default settings still apply for the other protocols.

- Remove any profile associations with UNP access ports before attempting to modify or delete the profile.

To delete a Layer 2 profile, use the **no** form of the **service l2profile** command. For example, the following command deletes the "DropL2" profile:

```
-> no service l2profile DropL2
```

Use the **show service l2profile** command to view a list of profiles that are already configured for the switch. This command also displays the attribute values for each profile. For example:

```
-> show service l2profile
Legend: (*) in-use by UNP
Profile Name: def-access-profile,
  STP     : tunnel,    802.1X  : drop,      802.3AD : peer,     802.1AB : drop,
  GVRP    : tunnel,    AMAP    : drop,      MVRP    : tunnel
Profile Name: DropL2,
  STP     : drop,      802.1X  : drop,      802.3AD : drop,     802.1AB : drop,
  GVRP    : drop,      AMAP    : drop,      MVRP    : tunnel
Profile Name: unp-def-access-profile*
  STP     : tunnel,    802.1X  : peer,      802.3AD : peer,     802.1AB  : drop,
  GVRP    : tunnel,    AMAP    : drop,      MVRP    : tunnel
```

Assigning Layer 2 Profiles to UNP Access Ports

After a Layer 2 profile is created, it is then necessary to assign the profile to a UNP access port or link aggregate. When this is done, the current profile associated with the port is replaced with the new profile.

The **unp l2-profile** command is used to assign a new profile to an access port. For example, the following commands assign the "DropL2" profile to UNP access port 1/4 and link aggregate 5:

```
-> unp port 1/4 l2-profile DropL2
-> unp port linkagg 5 l2-profile DropL2
```

To change the profile associated with the access port back to the default profile (**unp-def-access-profile**), specify the default profile name with the **unp l2-profile** command. For example:

```
-> unp port 1/4 l2-profile default
-> unp linkagg 5 l2-profile default
```

Use the **show unp port config** command to verify the Layer 2 profile assignment. For example:

```
-> show unp port 1/1/11 config
Port  1/1/11
  Port-Type                      = Access,
  802.1x authentication          = Enabled,
  802.1x Pass Alternate Profile  = -,
  802.1x Bypass                  = Disabled,
  802.1x failure-policy          = default,
  Mac-auth allow-eap             = -,
  Mac authentication             = Enabled,
  Mac Pass Alternate Profile     = -,
  Classification                 = Enabled,
  Trust-tag                      = Enabled,
  Default Profile                = -,
  Port Domain Num                = 0,
  AAA Profile                    = -,
  Port Template                  = accessDefaultPortTemplate,
  Admin State                    = Enabled,
  Dynamic Service                = spb,
  PVLAN Port Type                = -,
  Force L3-Learning              = Disabled,
  Force L3-Learning Port Bounce  = Enabled,
  802.1x Parameters:
          Tx-Period         = 30,
          Supp-Timeout      = 30,
          Max-req           = 2
  L2 Profile                = "unp-def-access-profile",
```

Configuring UNP for Silent Devices

A silent device connected to a UNP port may not receive the necessary broadcast packets to wake the device when dynamic UNP port assignments time out due to inactivity on the port. This section describes the following solutions to ensure traffic continues to flow to silent devices that are connected to UNP ports:

- Statically assigning a UNP service profile to a UNP port creates a persistent Service Access Point (SAP) that will not age out when there is no activity on the port. This solution is particularly useful for access to silent devices in the UNP service domain.
- Statically assigning a VLAN to a UNP port creates a VLAN-port association that will not age out when there is no activity on the port. This solution is particularly useful for access to silent devices in the UNP VLAN domain.

Statically Assigning Service Profiles for Silent Devices

When a MAC address is learned on a UNP port and classified into a service profile, a SAP is dynamically created based on the parameter values of the service profile. Once the MAC address associated with the dynamic SAP ages out, the SAP ages out as well. This poses a problem for silent devices connected to UNP access ports; when the device goes idle and the dynamic SAP ages out, the silent device no longer receives broadcast or multicast packets to wake the device.

To accommodate silent devices, assign a service profile to the UNP port. When the profile is assigned to the UNP port, a SAP is dynamically created based on the service parameter values defined for the profile. This action is automatically triggered even if a MAC address has not been learned on the port.

The SAP that is created when a service profile is assigned to a UNP port is a persistent SAP that will not age out when any MAC addresses learned on the SAP age out; the SAP continues to receive broadcast and multicast packets for the silent device even if there are no MAC addresses learned on the SAP.

Consider the following guidelines when statically assigning a service profile for silent devices:

- Make sure the specified UNP profile name already exists in the switch configuration and is mapped to an SPB, VXLAN, L2 GRE, or static service.
- Profiles mapped to SPB, VXLAN, or static services are configured as static profiles on UNP access ports.
- Profiles mapped to an L2 GRE service are configured as static profiles on UNP bridge or access ports.
- More than one SPB or VXLAN service profile can be statically assigned to the same UNP access port, but mixing service types on the same port is not supported. For example, configure only SPB service profiles or only VXLAN service profiles for the same access port.
- There can only be one L2 GRE service profile statically assigned to a UNP bridge or access port.

To assign a service profile to a UNP port, use the **unp port profile** command. For example, the following commands configure and assign service profile “static-spb1” to UNP access port 1/4/31:

```
-> unp profile static-spb1
-> unp profile static-spb1 map service spb tag-value 10 isid 1500 bvlan 500
-> unp port 1/4/31 port-type access
-> unp port 1/4/31 profile static-spb1
```

UNP service profile “static-spb1” is mapped to SPB service parameters. When this profile is assigned to UNP access port 1/4/31, a dynamic SPB SAP is automatically created to process traffic on that port. The 1/4/31 port SAP never ages out and is only taken down when the profile assignment is removed from the port.

To remove a profile assignment from a UNP port, use the **no** form of the **unp port profile** command. For example:

```
-> no unp port 1/4/31 profile static-spb1
```

Use the **show unp port profile** command to verify the UNP static profile configuration. For example:

```
-> show unp port profile
Port    Profile
-------+----------------
1/4/31   static-spb1
```

To verify that a dynamic service and SAP was created automatically when a service profile is assigned to a UNP port, use the **show service** and **show service ports** commands. For example:

```
-> show service
Legend: * denotes a dynamic object
All Service Info
           Svc                   SAP    Bind
ServiceId  Type  Adm  Oper Stats Count  Count  Description
----------+-----+----+----+-----+------+------+--------------------------------
32768*     SPB   Up   Down  N    1      0      Dynamic Service isid=1500 for UNP
```

```
-> show service 32768 ports
Legend: (*) dyn unicast object (+) remote mcast object (#) local mcast object
SPB Service 32768 (Dynamic Service isid=1500 for UNP)
Admin : Up,   Oper : Down,  Stats     : N,        Mtu   : 9194, VlanXlation : N,
ISID  : 1500, BVlan: 500,   MCast-Mode: Headend,  Tx/Rx : 0/0,  RemoveIngTag: N

                                Sap Trusted:Priority/        Sap Description /
Identifier      Adm  Oper Stats Sdp SystemId:BVlan   Intf    Sdp SystemName
---------------+----+----+-----+--------------------+-------+------------------
sap:1/4/31:10*  Up   Down  N           Y:x           1/4/31  Dynamic SAP for UNP
```

For more information about the commands described in this section, see the "Access Guardian Commands" chapter and the "Service Manager Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Statically Assigning VLANs for Silent Devices

When a MAC address is learned on a UNP bridge port and classified into a VLAN profile, a VLAN-port association is dynamically created between the port and the VLAN mapped to the profile. The UNP port becomes a member of that VLAN. However, when the MAC address ages out, the VLAN-port association also ages out and the UNP port is no longer a member of that VLAN. This is problematic for silent devices as they will no longer receive broadcast packets forwarded on the VLAN to wake the device.

To accommodate silent devices, statically assign a VLAN to the UNP bridge port. Doing so will automatically create a VLAN-port association between the port and VLAN that will not age out even if there are no MAC addresses learned on the port; the UNP bridge port continues to receive broadcast packets for any silent device that is connected to the port.

Consider the following guidelines when configuring a static VLAN for a UNP bridge port:

- Static VLANs are only configurable on UNP bridge ports (UNP access ports are not supported).
- Statically assigning a VLAN as an untagged or tagged VLAN for the UNP port is supported.
- When a VLAN is assigned to a UNP bridge port, the port goes into a forwarding state for egress traffic associated with the VLANs assigned to the port. This automatically occurs even when there is no MAC address learned on the UNP port in the assigned VLANs and regardless of the direction value (in or both) set for the port.

To configure an untagged or tagged VLAN assignment for a UNP bridge port, use the **unp vlan** command. For example, the following command assigns VLAN 100 as an untagged static VLAN assignment for UNP port 1/4/45:

```
-> unp port 1/4/45 vlan 100
```

To specify a tagged VLAN assignment, use the **tagged** parameter with the **unp vlan** command. For example:

```
-> unp port 1/4/45 vlan 100 tagged
```

Configuring a UNP port or link aggregate with an untagged *and* tagged VLAN-port association is allowed as long as the untagged and tagged VLANs are different. For example, the following commands configure an untagged and tagged VLAN assignment for the same UNP bridge port:

```
-> unp port 1/4/45 vlan 100
-> unp port 1/4/45 vlan 200 tagged
```

To remove a static VLAN assignment from a UNP port, use the **no** form of the **unp vlan** command. For example:

```
-> no unp port 1/4/45 vlan 100
```

Use the **show unp port configured-vlans** to display the static VLAN assignments for UNP bridge ports. For example:

```
-> show unp port configured-vlans
Port     Vlan   Type
-------+------+---------
0/10      500  unpUntag
0/10      501  unpUntag
1/1/10    600  unpQtag
1/1/11    601  unpUntag
1/1/11    602  unpQtag
1/1/11    603  unpQtag
```

For more information about the commands described in this section, see the "Access Guardian Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring UNP Profiles

A Universal Network Profile (UNP) is assigned to a host device through one of the following Access Guardian methods:

1 The device authentication process via a remote RADIUS-capable server, a Unified Policy Access Manager (UPAM) server, or a ClearPass Policy Manager (CPPM) server.

2 Application of profile classification rules, when authentication is not available or fails.

3 The UNP port configuration defines a default UNP profile for traffic that was not assigned to a profile through other Access Guardian methods.

To create a UNP profile, use the **unp profile** command. For example:

```
-> unp profile guest
-> unp profile employee
```

After a profile is created, configure the profile mapping to determine if a device is forwarded on the Access Guardian VLAN or service domain. Device traffic received on UNP bridge ports is eligible for assignment to VLAN-mapped profiles; device traffic received on UNP access ports is eligible for assignment to service-mapped profiles. For example, the following commands map a VLAN to the "guest" profile and map a service to the "employee" profile:

```
-> unp profile guest map vlan 200
-> unp profile employee map service-type spb tag-value 10 isid 1500 bvlan 500
```

Until a UNP profile is created and the VLAN or service mapping is configured, the profile is not available for Access Guardian assignment of devices connected to UNP ports. See "Configuring the UNP Profile Mapping" on page 30-62 for more information.

Configuring UNP Profile Attributes

When a profile is created with no other optional parameter values, the UNP profile attribute values listed in "Access Guardian Profile Defaults" on page 30-4 are applied to the new profile. To change the default UNP profile attribute values, use the commands listed in the following table:

Command	Description
unp profile qos-policy-list	Assigns a QoS policy list to a profile. If there is no list assigned to a profile, users classified into that profile are granted full access within the profile VLAN or service domain. See "Configuring QoS Policy Lists" on page 30-76.
unp profile location-policy	Assigns the name of a location-based policy to the profile. This type of policy defines criteria (such as the slot/port, system name, and location) to determine if a device is accessing the network from a valid location.
unp profile period-policy	Assigns the name of a time-based policy to the profile. This type of policy specifies the days and times during which a device can access the network.
unp profile captive-portal-authentication	Configures the status of internal Captive Portal authentication for the profile. When enabled, triggers the OmniSwitch Captive Portal authentication process for users classified into the profile.
unp profile captive-portal-profile	Assigns the name of a Captive Portal profile that applies a specific Captive Portal configuration to devices assigned to the UNP profile. This type of profile is applied when Captive Portal is enabled for the UNP profile and overrides the global Captive Portal configuration.
unp profile authentication-flag	Configures the status of the authentication flag for the profile. When enabled, only devices that were successfully authenticated are allowed into the profile.
unp profile mobile-tag	Configures the mobile tagging status for the profile. When enabled, the first user that is learned on a UNP port and classified into the UNP profile will cause the UNP port to be added as a tagged member of the VLAN associated with the profile. If the profile is mapped to a service, a tagged virtual port association is created.
unp profile maximum-ingress-bandwidth	Configures the maximum amount of bandwidth allocated for ingress traffic on UNP ports assigned to the profile.
unp profile maximum-egress-bandwidth	Configures the maximum amount of bandwidth allocated for egress traffic on UNP ports assigned to the profile.
unp profile maximum-ingress-depth	Configures how much traffic is allowed to burst over the maximum ingress bandwidth limits on UNP ports assigned to the profile.
unp profile maximum-egress-depth	Configures how much traffic is allowed to burst over the maximum egress bandwidth limits on UNP ports assigned to the profile.
unp profile inactivity-interval	Configures whether or not an authenticated device assigned to the profile is automatically logged out of the network after a specific period of inactivity (MAC address for the device has aged out).

unp profile mac-mobility	Configures the status of MAC address mobility for the profile. Enable this attribute when configuring VRRP to operate with dynamic UNP SAPs. See "VRRP over UNP Dynamic SPB SAPs" on page 30-32.
unp profile saa-profile	Assigns the name of a Service Assurance Agent (SAA) profile to the specified UNP profile. An SAA profile is mainly used by the OmniVista network management application to monitor connections between virtual machines (VMs) in a data center network. See "Configuring a Service Assurance Agent Profile" on page 30-66.

UNP Profile Configuration Guidelines

Consider the following guidelines when configuring UNP profile attributes:

- Any profile names that will be assigned through RADIUS authentication and/or the UPAM or CPPM BYOD process must be defined on the OmniSwitch and the RADIUS and/or UPAM or CPPM server.
- UNP profile attributes are only applied to device traffic that is received on UNP-enabled ports or link aggregates. See "Configuring UNP Port-Based Functionality" on page 30-41 for more information.
- The QoS rules within a policy list are applied to all members of the UNP profile group to enforce access to network resources. Only one policy list is allowed per profile, but multiple profiles may use the same policy list. See "Configuring QoS Policy Lists" on page 30-76 for more information.
- Specifying a QoS policy list name that is inactive or does not already exist in the switch configuration is allowed. However, the list will remain inactive for the UNP until the list is enabled or configured using the QoS policy list commands. See "Configuring QoS Policy Lists" on page 30-76 for more information.
- If a device violates a location or time period policy, the device is placed into an unauthorized state, even though it is still assigned to the UNP profile. In this state, a built-in QoS policy list is applied to the device to restrict the role of the device in the network. See "Built-in Restricted Roles" on page 30-15 for more information.
- Profile location and time period policies are configurable on the switch or on the RADIUS server. If the policies are configured on both the switch and the RADIUS server, then the switch policies take precedence.
- Captive Portal authentication is applied as a post-authentication and/or post-classification mechanism to devices assigned to the UNP profile. Captive Portal provides a Web-based authentication mechanism to dynamically change the role-based access (policy list) for a user. See "Using Captive Portal Authentication" on page 30-91 for more information.
- UNP profile redirection for BYOD is automatically made available to devices assigned to a VLAN-mapped profile based on the status of Captive Portal authentication for the profile:
  - When Captive Portal authentication is disabled (the default), BYOD redirection is automatically triggered when the initial device authentication process returns the "Alcatel-Redirect-URL" attribute. See "Bring Your Own Devices (BYOD) Overview" on page 30-150 for more information.
  - When Captive Portal authentication is enabled, internal Captive Portal is enforced and BYOD redirection is not available.
- To ensure proper BYOD redirection for devices classified into a UNP VLAN-mapped profile, configure the redirection server as the preferred server through AAA commands for MAC and 802.1X authentication. See "Setting Authentication Parameters for the Switch" on page 30-34 for more information.

- The maximum ingress bandwidth, egress bandwidth, and depth attribute values are applied to the port of a user device that is classified into the specified profile.
  - If multiple user devices are classified into different profiles but learned on the same UNP port, the profile bandwidth values that were applied for the last user learned are applied on the port. Parameter values applied through previously learned users are overwritten.
  - Bandwidth parameter values are *not* applied to UNP link aggregates that are assigned to the profile.
- UNP classification rules can be defined for a UNP profile to provide an additional method for assigning a device into a profile. If authentication is not available or does not return a profile name, classification rules are applied to determine the profile assignment. See "Configuring UNP Classification Rules" on page 30-80 for more information.
- A UNP profile can be configured as a default profile for a UNP port. If authentication and classification do not return a profile name, the device is then assigned to the default profile associated with the UNP port on which the device was learned. See "Configuring UNP Port Parameters" on page 30-42. for more information.

UNP profile attributes are configurable at the time a profile is created or for a profile that already exists. For example, the following command creates a new "guest" profile with a QoS policy list and enables the authentication flag and internal Captive Portal authentication:

```
-> unp profile guest qos-policy-list qlist1 authentication-flag captive-portal-
authentication
```

The next command example modifies the "guest" profile to disable the authentication flag:

```
-> unp profile guest no authentication-flag
```

The above command only changes the authentication flag status; the QoS policy list assignment and the internal Captive Portal status remain unchanged for the "guest" profile.

To verify the UNP profile configuration for the switch, use the **show unp profile** command. For example:

```
-> show unp profile guest
Profile Name: guest
    Qos Policy      = qlist1,
    Location Policy = loclist1,
    Period Policy   = timelist1,
    CP Profile      = guest-profile,
    CP State        = Ena,
    Authen Flag     = Dis,
    Mobile Tag      = Dis,
    SAA Profile     = -,
    Ingress BW      = -,
    Egress BW       = -,
    Ingress Depth   = -,
    Egress Depth    = -,
    Inact Interval  = 10
```

For more information about the commands described in this section, see the "Access Guardian Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring the UNP Profile Mapping

In addition to profile attributes, each profile is mapped to either a VLAN ID or to service-based parameters. The following types of services can be mapped to a UNP profile:

- Shortest Path Bridging (SPB)
- Virtual eXtensible LAN (VXLAN)
- Layer 2 Generic Routing Encapsulation (L2 GRE)
- A static service (the ID for an existing service is specified)

Only one type of profile mapping (VLAN, SPB, VXLAN, L2 GRE, or static) is associated with a profile at any given time. The type of mapping configured for a profile determines whether traffic received on UNP bridge ports or on UNP access ports is eligible for assignment to that profile. For example:

- Traffic received on UNP bridge ports is eligible for assignment to a VLAN profile.
- Traffic received on UNP access ports is eligible for assignment to a service profile.

This section describes how to configure the following different types of mappings for a UNP profile:

- "Mapping a VLAN to a UNP Profile" on page 30-62.
- "Mapping Service Parameters to a UNP Profile" on page 30-66.
- "Mapping a Static Service to a UNP Profile" on page 30-71.

For more information about profiles, see "UNP Profiles" on page 30-16.

Mapping a VLAN to a UNP Profile

The **unp profile map vlan** command is used to map a VLAN ID to a UNP profile. For example, the following command maps VLAN 400 to the "employee" profile:

```
-> unp profile employee map vlan 400
```

Devices classified into the "employee" profile are dynamically assigned to VLAN 400.

Consider the following when configuring a VLAN mapping for a UNP profile:

- The VLAN associated with a profile must already exist in the switch configuration, unless one of the following conditions occur:
  - The dynamic VLAN configuration functionality is enabled for the switch (see "Enabling Dynamic VLAN Configuration" on page 30-63).
  - The VLAN mapping to a profile is done when the switch boots up.
- Configuring a new VLAN mapping for a profile will overwrite the existing VLAN mapping for that profile. Any change to the mapping configuration of the profile will flush all MAC addresses learned on that profile.
- Removing a VLAN mapping configuration requires deleting the entire profile from the switch configuration (**no unp profile** *profile_name*).
- If a standard VLAN ID associated with a VLAN profile is deleted, the profile association with that VLAN ID is still maintained. Any traffic subsequently classified with this profile is filtered unless the UNP port on which the traffic is received is configured with alternate classification methods (see "Configuring UNP Port Parameters" on page 30-42).

To verify the VLAN profile configuration for the switch, use the **show unp profile map** command with the **vlan** parameter. For example:

```
-> show unp profile employee map vlan
Profile Name                     Vlan-Id
--------------------------------+--------
employee                         400

Total Profile Vlan-Map Count: 1
```

Enabling Dynamic VLAN Configuration

When creating a UNP VLAN profile, it is possible to specify the VLAN ID of a VLAN that does not exist in the switch configuration. The UNP feature provides the ability to enable dynamic VLAN configuration, which allows "on the fly" configuration of VLANs as they are needed.

When dynamic VLAN configuration is enabled and a profile is mapped to a VLAN that does not exist, UNP will create that VLAN at the time the profile mapping is created.

Dynamic VLAN configuration is a global UNP setting that applies to all VLAN profiles. By default, this setting is disabled for the switch. To enable this functionality, use the **unp dynamic-vlan-configuration** command.

```
-> unp dynamic-vlan-configuration
```

Use the **no** form of the **dynamic-vlan-configuration** command to disable dynamic VLAN configuration.

```
-> no unp dynamic-vlan-configuration
```

Consider the following when enabling dynamic VLAN configuration:

- The UNP dynamic VLAN can be mapped to any UNP profile.
- The status of the dynamic VLAN and other tagged port (non-UNP port) assignments are configurable using standard VLAN commands. In addition, the STP status of the VLAN is configurable and enabled by default when the dynamic VLAN is created.
- A dynamic VLAN cannot be deleted using standard VLAN commands (**no vlan** *vlan_id*).
- UNP dynamic VLANs are identified as a separate type of VLAN. The **show vlan** command will display this type with the default name of "UNP-DYN-VLAN" and the designated type as "UNP Dynamic Vlan". For example:

```
-> show vlan 450
Name                     : UNP-DYN-VLAN,
Type                     : UNP Dynamic Vlan,
Administrative State     : enabled,
Operational State        : disabled,
IP Router Port           : disabled,
IP MTU                   : 1500
```

- Dynamic VLANs are not saved in the "! VLAN:" section of the switch configuration file (**boot.cfg**). However, the **unp** commands to enable dynamic VLAN configuration and create the UNP are saved in the "! DA-UNP:" section of **boot.cfg** (see the following sample **boot.cfg** file). As a result, the VLAN is created again on the next switch bootup.

```
-> show configuration snapshot vlan
! VLAN:
vlan 1 admin-state enable
vlan 451 admin-state enable
```

```
vlan 777 admin-state enable
vlan 887-888 admin-state enable

-> show configuration snapshot da
! DA-UNP:
unp dynamic-vlan-configuration enable
unp profile "temp1"
unp profile "unpTemp"
unp profile "unpTemp2"
unp profile "temp1" map vlan 450
unp profile "unpTemp" map vlan 10
unp profile "unpTemp2" map vlan 10
unp classification mac-address 00:00:00:00:00:01 profile1 unpTemp2
unp classification mac-address 10:22:33:44:55:66 profile1 unpTemp2
unp classification ip-address 1.1.1.2 mask 255.0.0.0 profile1 unpTemp2
unp port 1/1/11 port-type bridge
unp port 1/1/11 802.1x-authentication
unp port 1/1/11 classification
unp port 1/1/12 port-type bridge
unp port 1/1/12 mac-authentication
unp port 1/1/12 classification
```

To verify the status of dynamic VLAN configuration for the switch, use the **show unp global configuration** command. For example:

```
-> show unp global configuration
  Dynamic Vlan Configuration     = Enabled,
  Dynamic Profile Configuration  = Disabled,
  Auth Server Down Profile1      = -,
  Auth Server Down Profile2      = -,
  Auth Server Down Profile3      = -,
  Auth Server Down Voice Profile1  = -,
  Auth Server Down Voice Profile2  = -,
  Auth Server Down Voice Profile3  = -,
  Auth Server Down Timeout       = 60,
  Redirect Port Bounce           = Enabled,
  Redirect Pause Timer           = -
  Redirect http proxy-port       = 8080
  Redirect Server FQDN           = cppm.abc.com
  Redirect Server IP             = 10.1.1.1
  Allowed IP                     = -
  Force L3-Learning              = Disabled
  Force L3-Learning Port Bounce  = Disabled
  802.1x Pass Through Mode       = Disabled
  AP Mode                        = Enabled
  System-default service-mod     = 512
  System-default service-base    = 10000000
  System-default Multicast-Mode  = Headend
  System-default Vlan-Xlation    = Enabled
  System-default Multicast-Group = 239.0.0.0
  System-default far-end-ip-list = -
  IPv6 Drop Packets              = Disabled,
  Delayed Learning Interval      = 0,
  Global Mac-Mobility            = Disabled,
```

Enabling Dynamic Profile Configuration

The UNP feature provides the ability to enable dynamic VLAN profile configuration, which allows "on the fly" configuration of profiles when specific traffic conditions occur. By default, dynamic profile configuration is disabled for the switch. To enable this functionality, use the **unp dynamic-profile-configuration** command.

```
-> unp dynamic-profile-configuration
```

Use the **no** form of the **dynamic-profile-configuration** command to disable this functionality.

```
-> no unp dynamic-profile-configuration
```

Dynamic profile configuration is a global UNP setting that is applied to traffic on any UNP bridge port that is configured to trust the VLAN tag of the incoming packets.

Consider the following when enabling dynamic profile configuration:

- A profile is only dynamically created if the trust VLAN tag is enabled for the UNP bridge port and the packet VLAN tag matches an MVRP VLAN ID that is not assigned to a UNP or there is no matching VLAN ID in the switch configuration.
- Dynamically created profiles are saved in the **boot.cfg** file for the switch.
- By default, dynamically created VLAN profiles are automatically named **dynamic_profile**_*vlan_id*, where the VLAN ID is the ID of the VLAN contained in the packet tag.
- After the dynamic profile is created, changing the VLAN profile name, associated VLAN ID, or the QoS policy list is allowed. To avoid any confusion, change the profile name if the VLAN ID associated with the profile has changed.
- When the dynamic profile configuration option is enabled along with the dynamic VLAN configuration option and the dynamically created profile refers to a VLAN that is an MVRP VLAN, then the MVRP VLAN is automatically converted to a dynamic UNP VLAN (UNP-DYN-VLAN).

To verify the status of dynamic profile configuration for the switch, use the **show unp global configuration** command. For example:

```
-> show unp global configuration
  Dynamic Vlan Configuration     = Enabled,
  Dynamic Profile Configuration  = Disabled,
  Auth Server Down Profile1      = -,
  Auth Server Down Profile2      = -,
  Auth Server Down Profile3      = -,
  Auth Server Down Voice Profile1  = -,
  Auth Server Down Voice Profile2  = -,
  Auth Server Down Voice Profile3  = -,
  Auth Server Down Timeout       = 60,
  Redirect Port Bounce           = Enabled,
  Redirect Pause Timer           = -
  Redirect http proxy-port       = 8080
  Redirect Server FQDN           = cppm.abc.com
  Redirect Server IP             = 10.1.1.1
  Allowed IP                     = -
  Force L3-Learning              = Disabled
  Force L3-Learning Port Bounce  = Disabled
  802.1x Pass Through Mode       = Disabled
  AP Mode                        = Enabled
  System-default service-mod     = 512
  System-default service-base    = 10000000
```

```
System-default Multicast-Mode  = Headend
System-default Vlan-Xlation    = Enabled
System-default Multicast-Group = 239.0.0.0
System-default far-end-ip-list = -
IPv6 Drop Packets              = Disabled,
Delayed Learning Interval      = 0,
Global Mac-Mobility            = Disabled,
```

Configuring a Service Assurance Agent Profile

A Service Assurance Agent (SAA) profile defines jitter and latency threshold values that are applied by SAA sessions to monitor the performance of network traffic associated with a UNP VLAN profile. An SAA profile is first created and then assigned to a UNP VLAN-based profile; UNP service-based profiles do not support this functionality.

Note. Although SAA profiles can be configured and assigned to a UNP through the CLI, these profiles are mainly used by the OmniVista network management application to monitor connections between virtual machines (VMs) in a data center network.

To configure an SAA profile, use the **unp saa-profile** command. For example, the following command creates an SAA profile named "unp_saa1" and defines both jitter and latency threshold values for the profile:

```
-> unp saa-profile unp_saa1 jitter-threshold 100 latency-threshold 500
```

To assign an SAA profile to a UNP VLAN profile, use the **unp profile saa-profile** command with the **saa-profile** parameter. For example, the following command assigns SAA profile "unp_saa1" to VLAN profile "unp1":

```
-> unp profile unp1 saa-profile unp_saa1
```

Mapping Service Parameters to a UNP Profile

There are three types of service-based mappings supported: Shortest Path Bridging (SPB), Virtual eXtensible LAN (VXLAN), and Layer 2 Generic Routing Encapsulation (L2 GRE).

When a device is dynamically assigned to an SPB, VXLAN, or L2 GRE service profile, a dynamic process is triggered to create an SPB, VXLAN, or L2 GRE Service Access Point (SAP) based on the service parameters specified in the profile mapping. Traffic from the device is then forwarded on the dynamically created SAP.

A SAP is comprised of the UNP access port on which device traffic is received, a VLAN tag value for the SAP encapsulation, and a service instance (SPB I-SID, VXLAN Network ID, L2 GRE tunnel ID). The encapsulation identifies the traffic received on the UNP access port that the SAP will forward on the service instance that is associated with the SAP.

Consider the following when mapping an SPB, VXLAN, or L2 GRE service to a UNP profile:

- Configuring a new service mapping for a profile will overwrite the existing service mapping for that profile. Any change to the mapping configuration of the profile will flush all MAC addresses learned on that profile.
- Removing a service mapping configuration requires deleting the entire profile from the switch configuration (**no unp profile** *profile_name*).

- The VLAN tag value indicates whether the VLAN tag information from the classified packets is used to assign the traffic to a SAP or if specific single or double-tagged values are used to assign the traffic to a SAP. Specify one of the following VLAN tag values for the profile:

0 (zero)	The VLAN tag of the packet is used to determine the SAP encapsulation value. For example, a SAP with an encapsulation value set to 1/12:5 is created when classified traffic received on UNP access port 1/12 is single-tagged with VLAN ID 5. Setting the profile tag value to zero has the same result as enabling trust VLAN tag for a UNP access port.
Outer VLAN tag	The outer VLAN tag value to use for the SAP encapsulation value.
Inner and outer VLAN tags	The inner and outer VLAN tag values to use for the SAP encapsulation value.

- If classified traffic is untagged, then zero is used for the SAP encapsulation value (for example, 1/2:0).
- If the VLAN tag value of the classified traffic does not match the tag value specified in the profile, UNP will check to see if the trust VLAN tag option is enabled for the UNP access port. If so, a SAP is assigned using the VLAN tag values of the traffic. If not, the traffic is learned as filtering on the UNP port.
- The egress VLAN translation status for the SPB, VXLAN, or L2 GRE service mapping associated with the profile is also configurable. By default, VLAN translation is disabled.
  - When enabled, the VLAN tags for profile traffic are processed according to the settings for the SAP on which the frames will egress, not according to the settings for the SAP on which the frames were received.
  - Enabling VLAN translation at the service level is only applicable if the corresponding access ports for the SAPs also have VLAN translation enabled.

Mapping Multiple UNP Profiles to the Same Service

It is possible to define a service mapping for multiple UNP profiles in which each profile is mapped to the same service instance but with different service parameter options. For example:

- UNP profile “UNP-1” is configured with an SPB service mapping that specifies I-SID 1500, BVLAN 500, VLAN tag 10, head-end multicast mode, VLAN translation enabled, and IGMP/MLD snooping disabled.
- UNP profile “UNP-2” is configured with an SPB service mapping that specifies I-SID 1500, BVLAN 600, VLAN tag 10, head-end multicast mode, VLAN translation enabled, and IGMP/MLD snooping disabled.

In this example, both “UNP-1” and “UNP-2” are configured with the same service instance (I-SID 1500) but the BVLAN ID is different. When users are classified into these profiles, a SAP is dynamically created based on the service parameters defined by the profile configuration and the user MAC address is forwarded on that SAP. However, if a user is reclassified into the other profile (“UNP-1” to “UNP-2”), the MAC address may get filtered.

Users classified into UNP service profiles can only have one service parameter configuration. So even though the same I-SID is used by both profiles (“UNP-1 and “UNP-2), one of the service parameter options is different (BVLAN 500 and BVLAN 600). This configuration mismatch may cause the MAC address of the user to be filtered when the user moves between “UNP-1” and “UNP-2”.

When a user MAC address is filtered due to this type of misconfiguration, change the UNP profile configuration so that the service parameter options match for both profiles. For example, "UNP-1" and "UNP-2" will both use BVLAN 500. Then when the user is reclassified into the other UNP profile, the MAC address will be forwarded and not filtered.

Note. Although an SPB service mapping for a UNP profile is used in the example, the same scenario can occur when a VXLAN or L2 GRE service mapping is configured for multiple UNP profiles.

Configuring an SPB Service Mapping

The following elements of an SPB service are mapped to a UNP profile:

- The name of an existing UNP profile.
- VLAN tag value.
- Service instance ID (I-SID).
- An existing Backbone VLAN (BVLAN) ID.
- Multicast Mode (head-end by default)
- VLAN Translation (disabled by default)
- IGMP and MLD Snooping (disabled by default)

To configure the mapping of SPB service parameters to an existing UNP profile, use the **unp profile map service-type spb** command. For example, the following command configures a service mapping for the "vNP1" profile that will dynamically create an SPB SAP to carry traffic tagged with VLAN 10 on SPB service instance 1500 and will bind the service instance to SPB backbone VLAN (BVLAN) 500:

```
-> unp profile vNP1 map service-type spb tag-value 10 isid 1500 bvlan 500
```

By default, the multicast replication mode is set to head-end for the SPB service mapping. To change this setting, use the **unp profile-map service-type spb** command with the optional **multicast-mode** parameter. For example, the following command changes the multicast replication mode to tandem for the service profile:

```
-> unp profile vNP1 map service-type spb tag-value 10 isid 1500 bvlan 500
multicast-mode tandem
```

By default, VLAN translation is disabled for the SPB service mapping. To change this setting, use the **unp profile-map service-type spb** command with the optional **vlan-xlation** parameter. For example, the following command enables egress VLAN translation for the service profile:

```
-> unp profile vNP1 map service-type spb tag-value 10 isid 1500 bvlan 500 vlan-
xlation
```

By default, IGMP and MLD snooping are disabled for the SPB service mapping. To change these settings, use the **unp profile-map service-type spb** command with the optional **igmp-snooping** or **mld-snooping** parameters (see "IPMS for UNP Dynamic SPB SAPs" on page 30-32). For example, the following command enables both IGMP and MLD snooping for the service profile:

```
-> unp profile vNP1 map service-type spb tag-value 10 isid 1500 bvlan 500 vlan-
xlation multicast-mode tandem igmp-snooping mld-snooping
```

In addition to configuring the IGMP and MLD snooping status, it is also possible to specify an IP Multicast Switching (IPMS) profile to use when IGMP or MLD snooping is enabled. An IPMS profile specifies a pre-defined configuration that can be applied to the global IPMS instance (all VLAN and

service instances) or to a specific VLAN or service instance. For example, the following command enables IGMP snooping for the UNP profile and specifies an IPMS profile name:

```
-> unp profile vNP1 map service-type spb tag-value 10 isid 1500 bvlan 500 vlan-
xlation multicast-mode headend igmp-snooping profile ipms-prof1
```

If an optional IPMS profile is not specified for the UNP profile mapping, the default IPMS configuration settings are applied for IGMP/MLD snooping of profile traffic.

To verify the SPB service profile configuration for the switch, use the **show unp profile map** command with the **service-type spb** parameter. For example:

```
-> show unp profile vNP1 map service-type spb
Profile                         Tag         Vlan    Mcast   Igmp  Igmp    Mld   Mld
Name                  Isid      Value BVlan Xlation  Mode   Snoop Profile Snoop Profile
---------------------+--------+-----+-----+-------+-------+-----+-------+-----+-------
unp1-spb              1500      10    400   Ena     Tandem  Ena   -       Ena   -
unp2-spb              1600      20:30 401   Ena     Headend Ena   ipms-2  Dis   -
unp3-spb              1700      10    500   Dis     Headend Dis   -       Dis   -
SystemDefault10000010 10000010 10    4000  Dis     Headend Dis   -       Dis   -

Total Profile Spb-Map Count: 4
```

Refer to the “Configuring Shortest Path Bridging” chapter in this guide for more detailed information about SPB services.

Configuring a VXLAN Service Mapping

The following elements of a VXLAN service are mapped to a UNP profile:

- The name of an existing UNP profile
- VLAN tag value
- VXLAN Network ID (VNID)
- A list of far-end IP addresses and/or a multicast group IP address to identify the VXLAN Tunnel End Points (VTEPs) for traffic classified into this profile.
- Multicast Mode (hybrid mode by default)
- VLAN Translation (disabled by default)

To configure the mapping of VXLAN service parameters to an existing UNP profile, use the **unp profile map service-type vxlan** command.For example, the following command creates the “vmCluster1” profile to assign device traffic tagged with VLAN 12 to VXLAN segment 100 which will participate in the 225.1.1.1 multicast group to tunnel traffic:

```
-> unp profile vmCluster1 map service-type vxlan tag-value 12 vnid 2300
multicast-group 225.1.1.1
```

The following command example creates the “vmCluster2” profile to assign device traffic tagged with VLAN 15 to VXLAN segment 150 which will participate in tunneling traffic to the VTEP IP addresses contained in the “vteps” far-end IP list:

```
-> unp profile vmCluster2 map service-type vxlan tag-value 15 vnid 2301 far-end-
ip-list vteps
```

By default, the multicast replication mode is set to hybrid for the VXLAN service mapping (traffic is tunneled from the service instance to both a group of VTEPs that belong to the same multicast group address *and* to the VTEP nodes that are not associated with the same multicast group address). To change

this setting, use the **unp profile-map service-type vxlan** command with the optional **multicast-mode** parameter. For example:

```
-> unp profile vmCluster2 map service-type vxlan tag-value 15 vnid 2301 far-end-
ip-list vteps multicast-mode headend
```

By default, egress VLAN translation is disabled for the VXLAN service mapping. To change this setting, use the **unp profile map service-type vxlan** command with the optional **vlan-xlation** parameter. For example:

```
-> unp profile vmCluster2 map service-type vxlan tag-value 15 vnid 2301 far-end-
ip-list vteps vlan-xlation
```

The **far-end-ip-list** parameter that is specified when creating a VXLAN service profile is the name of a list of IP addresses that was previously configured. These IP addresses identify the VTEPs to which VXLAN traffic classified into the VXLAN service is tunneled.

To create a list of VTEP IP addresses, use the **unp vxlan far-end-ip-list** command. For example, the following command creates the "vteps" list and adds IP addresses to that list:

```
-> unp vxlan far-end-ip-list vteps 10.1.1.1 20.1.1.1 30.1.1.1 40.1.1.1
```

To verify the VXLAN far-end IP address list configuration, use the **show unp vxlan far-end-ip-list** command. For example:

```
-> show unp vxlan far-end-ip-list toDataCenter2
Far-End-Ip-List Name: vteps, IP-Count: 4,
    IP-Addresses:
         10.1.1.1
         20.1.1.1
         30.1.1.1
         40.1.1.1
```

To verify the VXLAN service profile configuration for the switch, use the **show unp profile map** command with the **service-type vxlan** parameter. For example:

```
-> show unp profile map service-type vxlan
Profile                     Tag   Far-End-List Vlan    Mcast   Mcast     Mac
Name                  Vnid  Value              Xlation Mode    Group     Orchest
---------------------+-----+-----+------------+-------+-------+---------+-------
vmCluster1            2300  12    -             Ena    Tandem  225.1.1.1  Dis
vmCluster2            2301  15    vteps         Ena    Headend -          Dis

Total Profile Vxlan-Map Count: 2
```

Refer to the "Configuring a VXLAN Gateway" chapter in the *OmniSwitch AOS Release 8 Data Center Switching Guide* for more detailed information about VXLAN services.

Configuring an L2 GRE Service Mapping

The following elements of an L2 GRE service are mapped to a UNP profile:

- The name of an existing UNP profile
- VLAN tag value
- An L2 GRE tunnel ID (VPNID)
- An IP address to identify an L2 GRE tunnel end point on which traffic classified into this profile is forwarded.

- VLAN translation (disabled by default)

L2 GRE services do not support multicast modes. By default, all Broadcast, Unknown Unicast, and Multicast (BUM) traffic is replicated by sending a copy to each far-end node over unicast SDPs. This is similar to how the head-end multicast mode works.

To configure the mapping of L2 GRE service parameters to an existing UNP profile, use the **unp profile map service-type l2gre** command. For example, the following command creates the "guest-profile" profile to assign device traffic tagged with VLAN 12 to VPNID 100 which will participate in tunneling traffic to the IP address of the far-end tunnel aggregation switch:

```
-> unp profile guest-profile map service-type l2gre tag-value 12 vpnid 100 far-
end-ip 20.2.2.1 vlan-xlation
```

To verify the L2 GRE service profile configuration for the switch, use the **show unp profile map** command with the **service-type l2gre** parameter. For example:

```
-> show unp profile map service-type l2gre
Profile                    Tag      Far-End-List   Far-End-Ip Port       Vlan
Name                Vpnid  Value                               Isolation  Xlation
-------------------+------+--------+--------------+----------+----------+-------
guest-profile       100    12                      20.2.2.1   Ena        Dis

Total Profile L2gre-Map Count: 1
```

Configuring a UNP profile mapped to an L2 GRE service identifies the switch as a tunnel access endpoint (the point at which traffic enters the L2 GRE tunnel). Additional configuration is required to define other L2 GRE tunnel endpoints. Refer to "Using L2 GRE Tunneling" on page 30-113 for more detailed information about L2 GRE services.

Mapping a Static Service to a UNP Profile

To configure the mapping of an existing SPB, VXLAN, or L2 GRE service to an existing UNP profile, use the **unp profile map service-type static** command. For example, the following command configures a static service mapping for the "unp1-staticSPB" profile that will carry traffic tagged with VLAN 20 on the existing SAP that is associated with service ID 10:

```
-> unp profile unp1-staticSPB map service-type static tag-value 20 service-id 10
```

To verify the static service profile configuration for the switch, use the **show unp profile map** command with the **service-type static** parameter. For example:

```
-> show unp profile map service-type static
Profile                                Tag
Name                            SvcId  Value
--------------------------------+------+-------
unp1-staticSPB                  10     20
unp2-staticVXLAN                20     40:50

Total Profile Static-Service-Map Count: 2
```

Setting the RADIUS Server Attribute Precedence

The Filter ID and Tunnel Private Group ID attribute values can be used to determine the UNP profile assignment for a device. Setting the attribute precedence determines which of these two values are checked first to obtain the UNP profile name for device authentication. For example, if the attribute precedence is set to Filter ID (the default) and the RADIUS server returns values for both attributes, then the Filter ID value is used to determine the UNP profile assignment.

To change the attribute precedence, use the **aaa radius unp-profile-precedence** command. For example:

```
-> aaa radius unp-profile-precedence tunnel-private-group-id
-> aaa radius unp-profile-precedence filter-id
```

Example Configuration for Setting the Attribute Precedence

```
-> vlan 1 admin-state enable
-> vlan 20 admin-state enable
-> vlan 20 members port 1/1/1 tagged

-> unp profile "20"
-> unp profile "20" map vlan 20
-> unp port-template pt direction both aaa-profile "rad1" classification trust-
tag admin-state enable
-> unp port-template pt mac-authentication
-> unp port 1/1/23 port-type bridge
-> unp port 1/1/23 port-template pt

-> aaa radius-server "rad1" host 192.168.20.100 key radius-key
-> aaa profile "rad1"
-> aaa profile "rad1" device-authentication mac "rad1"
-> aaa profile "rad1" radius mac-format username delimiter none case lowercase
-> aaa profile "rad1" radius mac-format password delimiter none case lowercase
```

Example 1	Device classified into Profile 20 using Filter-Id attribute.
RADIUS Configuration	0123456789ab - Cleartext-Password := "0123456789ab" - Filter-Id = "20"

```
-> show unp user
                                   User
Port    Username            Mac address       IP   Vlan Profile  Type    Status
-----+-------------------+-----------------+---+-----+-------+------+-----------
1/1/23 01:23:45:67:89:ab  01:23:45:67:89:ab -   20    20      Bridge Active
Total users : 1
```

Example 2	Device classified into Profile 20 using Tunnel-Private-Group-Id attribute.
RADIUS Configuration	0123456789ab - Cleartext-Password := "0123456789ab" - Tunnel-Type = VLAN, - Tunnel-Medium-Type = IEEE-802, - Tunnel-Private-Group-Id = 20

```
-> show unp user
                                   User
Port    Username            Mac address       IP   Vlan Profile  Type    Status
-----+-------------------+-----------------+---+-----+-------+------+-----------
1/1/23  01:23:45:67:89:ab 01:23:45:67:89:ab -   20    20      Bridge Active
Total users : 1
```

Example 3	Device classified into Profile 20 using Tunnel-Private-Group-Id with a higher precedence.
RADIUS Configuration	0123456789ab - Cleartext-Password := "0123456789ab" - Filter-Id = "incorrect", - Tunnel-Type = VLAN, - Tunnel-Medium-Type = IEEE-802, - Tunnel-Private-Group-Id = 20

```
-> aaa radius unp-profile-precedence tunnel-private-group-id
-> show unp user
                                User
Port    Username            Mac address      IP   Vlan Profile  Type    Status
-----+-------------------+-----------------+---+-----+-------+------+-----------
1/1/23 01:23:45:67:89:ab  01:23:45:67:89:ab -   20    20      Bridge Active
Total users : 1
```

Example 4	Device blocked since higher precedence Filter-Id attribute “incorrect” doesn’t exist.
RADIUS Configuration	0123456789ab - Cleartext-Password := "0123456789ab" - Filter-Id = "incorrect", - Tunnel-Type = VLAN, - Tunnel-Medium-Type = IEEE-802, - Tunnel-Private-Group-Id = 20

```
-> aaa radius unp-profile-precedence filter-id
-> show unp user
                                User
Port    Username            Mac address      IP   Vlan Profile  Type    Status
-----+-------------------+-----------------+---+-----+-------+------+-----------
1/1/23  01:23:45:67:89:ab 01:23:45:67:89:ab -    1     -      Bridge Block
Total users : 1
```

Configuring System Default Profile Parameters

System Default profiles are dynamically created to accommodate device traffic received on UNP access ports that is not classified into a user-defined UNP service profile. Parameters defined for the System Default profile will generate either an SPB SAP or VXLAN SAP based on the dynamic service setting configured for the UNP port on which the traffic is received. For more information, see "System Default Profiles" on page 30-20.

One of the parameter values defined for a System Default profile is an SPB I-SID or VXLAN VNID. This value is dynamically calculated using a base service number (10,000,000), modulo number (512), the VLAN tag of the UNP port traffic, and the UNP port domain value. For example, if the base service number is 10,000,000, the modulo number is 512, the VLAN tag is 30, and the domain is 10, then the following calculation is used to determine the SPB I-SID or VXLAN VNID number:

10000000 + (10 * 10000) + (30 % 512) = 10100030

The result of this example calculation, "10100030", is used as the SPB I-SID or VXLAN VNID parameter value for a System Default profile. If a different value is required, the base service number and modulo number can be changed to alter the result of the calculation.

Consider the following when configuring the base service and modulo values:

- Only the base service number and modulo number values are configurable; these values are set on a global basis. The calculation to determine the SPB I-SID or VXLAN VNID parameter values for all System Default profiles will use the same base service number and module number.
- When the base service number value is changed, subsequent System Default profiles are created with the new value while profiles created with the previous base value are retained.
- When the modulo number value is changed, all users learned in System Default profiles are flushed (logged out of the network) and dynamic SAPs created for the profiles are cleared.

To change the base service instance number, use the **unp system-default service-base** command. For example:

```
-> unp system-default service-base 5000
```

To set the base service number back to 10,000,000, use the **unp system-default service-base** command with the **default** parameter. For example:

```
-> unp system-default service-base default
```

To change the modulo number, use the **unp system-default service-mod** command. For example:

```
-> unp system-default service-mod 800
```

To set the modulo number back to 512, use the **unp system-default service-mod** command with the **default** parameter. For example:

```
-> unp system-default service-mod default
```

Configuring Dynamic Service Parameters

An SPB or VXLAN service is dynamically created for an SPB or VXLAN System Default profile. A reserved service ID number (32768) is assigned to a dynamic service; this number is incremented by 1 for each additional dynamic service (SPB or VXLAN) that is created and only has local significance.

The multicast mode and VLAN translation status is configurable for both SPB and VXLAN dynamic services. In addition, the multicast group IP address and far-end IP list name is configurable for VXLAN

dynamic services. For more information, see "System Default Profiles" on page 30-20.

Dynamic service parameter values are configured on a global basis. The switch will use the specified parameter values when dynamically creating a service for a System Default profile.

To change the multicast mode setting for dynamic SPB and VXLAN services, use the **unp system-default multicastmode** command. For example:

```
-> unp system-default multicast-mode tandem
-> unp system-default multicast-mode headend
```

To change the VLAN translation status for dynamic SPB and VXLAN services, use the **unp system-default vlan-xlation** command. For example:

```
-> unp system-default vlan-xlation disable
-> unp system-default vlan-xlation enable
```

To specify a multicast group IP address for dynamic VXLAN services, use the **unp system-default multicastgroup** command. For example:

```
-> unp system-default multicast-group 225.1.1.2
-> unp system-default multicast-group 239.0.0.0
```

To specify the name of a far-end IP list to associate with dynamic VXLAN services, use the **unp system-default far-end-ip-list** command. For example:

```
-> unp system-default far-end-ip-list vtep-list1
```

Verifying the System Default Profile Configuration

Use the **show unp global configuration** command to display the current settings for the global System Default profile parameters. For example:

```
-> show unp global configuration
  Dynamic Vlan Configuration     = Enabled,
  Dynamic Profile Configuration  = Disabled,
  Auth Server Down Profile1      = -,
  Auth Server Down Profile2      = -,
  Auth Server Down Profile3      = -,
  Auth Server Down Voice Profile1  = -,
  Auth Server Down Voice Profile2  = -,
  Auth Server Down Voice Profile3  = -,
  Auth Server Down Timeout       = 60,
  Redirect Port Bounce           = Enabled,
  Redirect Pause Timer           = -
  Redirect http proxy-port       = 8080
  Redirect Server FQDN           = cppm.abc.com
  Redirect Server IP             = 10.1.1.1
  Allowed IP                     = -
  Force L3-Learning              = Disabled
  Force L3-Learning Port Bounce  = Disabled
  802.1x Pass Through Mode       = Disabled
  AP Mode                        = Enabled
  System-default service-mod     = 512
  System-default service-base    = 10000000
  System-default Multicast-Mode  = Headend
  System-default Vlan-Xlation    = Enabled
  System-default Multicast-Group = 239.0.0.0
  System-default far-end-ip-list = -
```

```
  IPv6 Drop Packets              = Disabled,
  Delayed Learning Interval      = 0,
  Global Mac-Mobility            = Disabled,
```

The **show unp profile map** command will also display System Default profile information. For example:

```
-> show unp profile map service-type spb
Profile                                   Tag                Vlan     Mcast
Name                             Isid     Value    BVlan     Xlation  Mode
--------------------------------+--------+--------+--------+--------+---------
unp1-spb                         1500     10       400      Ena      Tandem
unp2-spb                         1600     20:30    401      Ena      Headend
SystemDefault10100030            10000010 10       4000     Dis      Headend

Total Profile Spb-Map Count: 3

-> show unp profile map service-type vxlan
Profile                              Tag   Far-End-List Vlan    Mcast   Mcast
Name                         Vnid    Value              Xlation Mode    Group
-------------------------+--------+-----+------------+-------+-------+---------
unp1-vxlan                2300     20    vtep-ip1     Ena     Tandem  225.1.1.2
unp2-vxlan                2301     40:50 vtep-ip2     Ena     Headend -
SystemDefault10000010     10000010 10    vtep-ip3     Dis     Headend -

Total Profile Vxlan-Map Count: 3
```

Configuring QoS Policy Lists

One of the attributes for UNP profiles specifies the name of a list of QoS policy rules. This list is applied to a user device when the device is initially assigned to the profile. Using policy lists allows the administrator to associate a group of users to a set of QoS policy rules. The policy rules applied determine the initial role (network access) for a user device classified into the profile.

To create a QoS policy list to assign to a UNP profile, use the **policy list** command to specify a list name and then use the **policy list rules** command to specify the names of one or more existing QoS/ACL policy rules to add to the list. For example, the following commands create two policy rules and associates these rules with the “temp-rules” list:

```
-> policy condition c1 802.1p 5
-> policy action a1 disposition drop
-> policy rule r1 condition c1 action a1
-> policy condition c2 source ip 10.5.5.0
-> policy action a2 disposition accept
-> policy rule r2 condition c2 action a2
-> policy list temp_rules type unp
-> policy list temp_rules rules r1 r2
-> qos apply
```

The following command example uses the **unp profile qos-policy-list** command to assign the “temp_rules” list to the “guest_user” UNP profile:

```
-> unp profile guest_user qos-policy-list temp_rules
```

Note the following guidelines when configuring QoS policy rules and lists:

- A default policy list exists in the switch configuration. Rules are added to this list when the rule is created. A rule can belong to multiple policy lists. As a result, the rule remains a member a of the default list even when it is subsequently assigned to additional lists.

- Each time a rule is assigned to a policy list, an instance of that rule is created. Each instance is allocated system resources. To exclude a rule from the default policy list, use the **no default-list** option of the **policy rule** command when the rule is created. For example:

```
-> policy rule r1 condition c1 action a1 no default-list
```

- Up to 32 policy lists (including the default list) are supported per switch. Only one policy list per UNP is allowed, but a policy list can be associated with multiple profiles.
- If a rule is a member of multiple policy lists but one or more of these lists are disabled, the rule is still active for those lists that are enabled.
- If the QoS status of an individual rule is disabled, then the rule is disabled for all policy lists, even if a list to which the policy belongs is enabled.
- Policy lists are not active on the switch until the **qos apply** command is issued.
- QoS policy lists that contain rules with a link aggregate source port condition are not supported on the OmniSwitch 6560 or the OmniSwitch 9900. In addition, using policy lists that contain rules with a source port condition are not supported when applied to 10G ports on an OmniSwitch 6560.
- On the OmniSwitch 6465, policy rules containing the following conditions are not supported in a UNP policy list:
  - Source port group
  - Source IPv6 address
  - IPv6 next header
  - IPv6 flow label
- On the OmniSwitch 9900, only policy rules with the following conditions can be assigned to a UNP policy list:
  - Destination MAC
  - EtherType / IPv6 Hop limit
  - Source VLAN
  - SIP
  - DIP / DIPv6
  - Layer 4 Protocol /NextHeader
  - Layer 4 source port
  - Layer 4 destination port
  - Source port bitmap

Use the **show policy list** command to display the QoS policy rule configuration. For example:

```
-> show policy list temp_rules
 Group Name                        From  Type    Enabled  Entries
 ---------------------------------+-----+-------+--------+----------------------
 temp_rules                         cli   unp     Yes      r1
                                                           r2
```

Dynamically Changing the Policy List Assignment (User Role)

The QoS policy list assigned to a UNP profile determines the initial role (network access) for a user device classified into the profile. This role can be dynamically changed for the user through the Captive Portal authentication mechanism, when a different policy list is returned for the user from a RADIUS, Unified Policy Access Manager (UPAM), or ClearPass Policy Manager (CPPM) server, or when the user is placed into a Captive Portal pre-login, unauthorized, or quarantined state.

Configuring an Explicit Policy List

When the switch assigns a user device to one of the restricted role states (unauthorized, Quarantine Manager, or Captive Portal pre-login), a built-in policy list associated with the restricted role is applied to the user. To override the built-in policy list with an explicitly configured policy list, use the **unp restricted-role policy-list** command. For example:

```
-> unp restricted-role unauthorized policy-list unauth1
-> unp restricted-role qmr policy-list quarantined1
-> unp restricted-role cp-prelogin policy-list cplogin1
```

When an explicit policy list assignment is removed, the switch reverts back to using the built-in policy list that is associated with the restricted role state.

Use the **show unp restricted-role** command to display the explicit policy list configuration for restricted roles. For example:

```
-> show unp restricted-role
Role name    Qos Policy List Name
------------+------------------------
UNAUTHORIZED qlist-bad
QMR          qlist-qmr
CP PRE-LOGIN qlist-cp

Total Restricted Role Count: 3
```

Configuring a User-defined Role

A user-defined role is used to define a list of conditions that a device must match and a QoS policy list name that is applied to devices matching the specified conditions. When the current context of a user device matches all of the role conditions, then the policy list associated with the role is applied to the device.

Only one user-defined role per user is allowed because only one QoS policy list per user is allowed. However, every time the user context changes for a device, all the user-defined roles are checked to see if there is a role that matches the current user context.

A user-defined role consists of the following components:

- A role name.
- A precedence value used to determine precedence among other user-defined rules. The valid precedence range is 1 (lowest) through 255 (highest).
- One or more of the following conditions:
  - The name of a UNP profile to which the user must belong.
  - The device is not authenticated.
  - The type of authentication (802.1X or MAC) the device successfully passed or failed.
  - The device is in a Captive Portal post-login state.

- The name of an existing QoS policy list.

To configure a user-defined role, use the **unp user-role** command. For example:

```
-> unp user-role role1 precedence 10
-> unp user-role role1 policy-list role1-list
-> unp user-role role1 profile1 unp1-vlan profile2 unp2-spb
-> unp user-role role1 authentication-type 8021x
-> unp user-role role1 cp-status-post-login
```

Use the **show unp user-role** command to display the user-defined role configuration. For example:

```
-> show unp user-role
Role Name: role1
  Qos Policy List       : role1-list
  Priority              : 10
  Conditions:
    Profile1            : unp1-vlan
    Profile2            : unp2-spb
    Profile3            : -
    Authentication-Type : 802.1x
    CP Status           : Enabled

Role Name: role2
  Qos Policy List       : qlist-allow
  Priority              : 1
  Conditions:
    Profile1            : -
    Profile2            : -
    Profile3            : -
    Authentication-Type : 802.1x Fail
    CP Status           : Disabled

Total User Role Derivation Rule Count: 2
```

Configuring UNP Classification Rules

UNP classification rules are defined and associated with UNP profiles to provide an additional method for classifying a device into a profile. If authentication is not available or does not return a profile name for whatever reason, classification rules are applied to determine the profile assignment.

The following table lists the classification rules that are supported and the **unp classification** command that is used to configure each rule:

Precedence Step/Rule	Command
1. Port	**unp classification port**
2. Domain ID	**unp classification domain**
3. MAC Address	**unp classification mac-address**
4. MAC OUI	**unp classification mac-oui**
5. MAC Address Range	**unp classification mac-range**
6. LLDP Media Endpoint Devices	**unp classification lldp med-endpoint**
7. Authentication Type	**unp classification authentication-type**
8. IP Address	**unp classification ip-address**
9. VLAN Tag	**unp classification vlan-tag**

For example, the following command is used to configure a MAC address range rule and assign that rule to an existing UNP profile named “Engineering”:

```
-> unp classification mac-address-range 00:11:22:33:44:55 00:11:22:33:44:66
profile1 Engineering
```

If the source MAC address of a device falls within the specified range of the example rule, then the device is classified into the “Engineering” profile and assigned to the VLAN or service associated with that profile.

Use the **show unp classification** command to verify the UNP classification rule configuration for the switch. For example, the following command displays the MAC address range rule configuration:

```
-> show unp classification mac-range-rule

Low MAC Address    High MAC Address   VLAN Tag Profile1 Name Profile2 Name Profile3 Name
------------------+------------------+--------+-------------+-------------+-------------
00:11:22:33:44:66  00:11:22:33:44:77  -        Engineering   -             -
00:11:22:33:44:88  00:11:22:33:44:99  10       CustB         VNP-B         -

Total Mac Range Rule Count: 2
```

For more information about UNP rules, see “UNP Classification Rules” on page 30-24.

Configuring the VLAN Tag Classification Rule

There are two methods for configuring classification rules that UNP will apply to device traffic that is tagged with a specific VLAN ID:

- Use the **unp classification vlan-tag** command to configure a VLAN ID tag rule that is applied only to traffic that is tagged with the specified VLAN ID. For example, the following command creates a VLAN tag rule that will assign traffic tagged with VLAN 10 to the “serverA” profile:

```
-> unp classification vlan-tag 10 profile1 serverA
```

- Combine the VLAN ID tag rule with other rules to include the tag as a required parameter to match for the rule. For example, to include the VLAN tag with a MAC address rule, use the **unp classification mac-address rule** command with the **vlan-tag** option:

  ```
  -> unp classification mac-address 00:00:2a:33:44:01 vlan-tag 10 profile1 serverA
  ```

  In this example, a device is classified into UNP "serverA" profile if the source MAC address of the device is "00:00:2a:33:44:01" *and* device packets are tagged with VLAN 10.

When a VLAN tag rule is combined with another rule, the combined rule takes precedence over the rule that does not specify a VLAN tag. For example, a rule that specifies a MAC address *and* a VLAN tag takes precedence over a rule that specifies only a MAC address.

Configuring the Domain Classification Rule

An optional UNP domain ID is assigned to UNP ports to form a logical group of ports to which classification rules are applied. There are two methods for configuring classification rules to apply to traffic received on ports in a specific domain ID:

- Use the **unp classification domain** command to configure a domain ID rule that is applied only to ports that belong to the specified domain ID. For example, the following command configures a domain rule that will classify device traffic into the "serverB" profile if the device is connected to a UNP port that is assigned to domain 2:

  ```
  -> unp classification domain 2 profile1 serverB
  ```

- Combine the domain classification rule with other rules to include the domain ID as a required parameter to match for the rule. For example, to include the domain ID with a MAC address rule, use the **unp classification mac-address rule** command with the **domain** option:

  ```
  -> unp classification mac-address 00:00:2a:33:44:01 domain 2 profile1 serverB
  ```

  In this example, device traffic is classified into the "serverB" profile if the source MAC address of the device is "00:00:2a:33:44:01" *and* the device is connected to a UNP port that is assigned to UNP domain 2.

The domain ID specified in a classification rule must already exist in the switch configuration. See "Configuring UNP Port Domains" on page 30-52 for more information.

Configuring the LLDP MED Endpoint Classification Rule

There are two types of configurable LLDP MED Endpoint rules: one for detecting IP phone traffic and one for detecting OmniAccess Stellar access point (AP) traffic.

- Use the **unp classification lldp med-endpoint** command with the **ip-phone** option to configure a rule that will detect LLDP TLVs from IP phones and then classify the traffic from the phones into the profile associated with the rule. For example:

  ```
  -> unp classification lldp med-endpoint ip-phone profile1 unp1-vlan
  ```

- Use the **unp classification lldp med-endpoint** command with the **access-point** option to configure a rule that will detect LLDP TLVs from Stellar APs and then classify the traffic from the APs into the profile associated with the rule. For example:

```
-> unp classification lldp med-endpoint access-point profile1 defaultWLANProfile
```

Note. An LLDP MED Endpoint AP rule is implicitly created and assigned to "defaultWLANProfile" (a built-in UNP profile on the switch) when the switch boots up. This facilitates the automatic discovery and management of OmniAccess Stellar APs that are connected to the switch.

Configuring Binding Rules for UNP Profiles

A binding rule defines a combination of one or more individual rules, all of which a device has to match. The following binding rule combinations are configurable and are listed in the order of precedence:

1. Port + MAC address + IP address
2. Port + MAC address
3. Port + IP address
4. Domain ID + MAC address + IP address

The precedence order of binding rules is used to determine precedence among only binding classification rules. However, all binding rules take precedence over all individual rules. So if a device matches both an individual rule and a binding rule, the device is classified into the profile associated with the binding rule.

The same commands used to configure individual classification rules are also used to configure binding rule combinations. For example, the **unp classification mac-address** command is used in the following example to configure a binding rule that combines a MAC address rule, an IP address rule, and a port rule:

```
-> unp classification mac-address 00:11:22:33:44:55 ip-address 10.0.0.20 mask
255.255.0.0 port 1/1/1 profile1 serverA
```

If the source MAC address, source IP address, *and* port of a device matches the MAC address, IP address, and port defined in the example binding rule, then the device is classified into the "serverA" profile and assigned to the VLAN associated with that profile.

Configuring Extended Classification Rules for UNP Profiles

An Extended classification rule defines a list of individual rules and assigns the list a name and a precedence value. A device must match all of the rules specified in the extended rule list.

The **unp classification-rule** command is used to create an extended rule and set the precedence value for the rule. The following commands are used to define classification rules and assign the rules to the extended rule name:

Precedence Step/Rule	Command
1. Port	**unp classification-rule port**
2. Domain ID	**unp classification-rule domain**
3. MAC address	**unp classification-rule mac-address**
4. MAC OUI	**unp classification-rule mac-oui**
5. MAC address range	**unp classification-rule mac-range**
6. LLDP Media Endpoint Devices	**unp classification-rule lldp med-endpoint**
7. Authentication Type	**unp classification-rule authentication-type**
8. IP address	**unp classification-rule ip-address**
9. VLAN tag	**unp classification-rule vlan-tag**

For example, the following commands create an extended classification rule named “ext-r1” with the precedence value set to 255 and assign the rule to a the “corporate” UNP profile:

```
-> unp classification-rule ext-r1 precedence 255
-> unp classification-rule ext-r1 profile1 corporate
```

Next, the following commands define a port rule and an authentication type rule and assign the rules to the “ext-r1” extended rule:

```
-> unp classification-rule ext-r1 port 1/1/10
-> unp classification-rule ext-r1 authentication-type 8021x
```

Note that the “ext-1” rule combines a port rule and an authentication type rule. This combination of rules is not allowed in a binding rule configuration.

The precedence value assigned to an extended classification rule is used to determine precedence only among extended classification rules. However, all extended rules take precedence over all individual and all binding rules. So if a device matches a binding rule (or an individual rule) and an extended rule, the device is classified into the profile associated with the extended rule.

Use the **show unp classification-rule** command to verify the UNP extended classification rule configuration for the switch. For example:

```
-> show unp classification-rule
Rule Name: "r1"
  Precedence              = 255,
  Profile1                = corporate,
    Conditions:
      Domain              = 0,
      Port                = 1/1/10,
      Authentication-Type = 802.1x,
Rule Name: "ext_r2"
  Precedence              = 1,
  Profile1                = unp1-vlan,
  Profile2                = unp2-vxlan,
    Conditions:
      Domain              = 20,
      Mac-Address         = 00:2a:94:11:22:01,
      Port                = 1/1/9,
      LLDP MED Endpoint   = IP-Phone,
      Authentication-Type = None,
```

Using Router Domain Authentication

Router domain authentication is used to authenticate users that try to access specific secure IPv4 destination networks. This is particularly useful for granting access to authorized administrators operating in IoT networks across router domains where end user MAC addresses are not available. An authorized user may have authenticated in another network but requires access to a different secure network; only the device IP address is known in this case, not the MAC address.

A user attempting to access a secure domain is challenged with either Captive Portal authentication or IP-based authentication to allow or deny access based on the type of traffic the user is sending.

- Users sending HTTP/HTTPS traffic are challenged with Captive Portal authentication.
- Users sending IP-based traffic are challenged with IP address based authentication.

Configuration Overview and Guidelines

Configuring router domain authentication requires the following steps:

1 Assign specific IP networks to a UNP network group. This type of group is used to identify secure networks; the group name is then used as a condition for triggering router domain authentication.

2 Create a UNP authentication user group that specifies a network group as a source or destination network condition.

3 Optionally assign a Captive Portal profile to specify authentication session parameters that are applied to user traffic that matches the user group condition. If a Captive Portal profile is not used, then the global Captive Portal configuration for the switch is applied.

The following sections provide information and guidelines for using router domain authentication:

- "UNP Router Authentication Guidelines" on page 30-85.
- "UNP Router Authentication User Group Guidelines" on page 30-85.
- "Captive Portal Profile Guidelines" on page 30-85.
- "Configuring Router Authentication" on page 30-86.
- "Router Domain Authentication Example" on page 30-89.

UNP Router Authentication Guidelines

Consider the following guidelines when setting up router domain authentication:

- Do not include the following IP networks when defining a UNP network group:
  - The Captive Portal IP subnet. This subnet is used exclusively by the internal Captive Portal feature to redirect DNS requests to the Captive Portal login screen.
  - The IP address of the DNS server.
- Make sure the Captive Portal IP subnet address is reachable to the client that is trying to access any of the destination networks.
- Any HTTP/HTTPS traffic sent to an IP address defined in a destination network group that goes through a proxy server will be challenged for authentication only if the proxy server is also included in a router authentication user group configuration.
- A Captive Portal logout request from a user (a source IP) learned in the system will terminate all the active sessions for that user.
- When a user login session ends (session timeout expires), the user is once again challenged for authentication. However, if a session timeout value of zero is returned from the RADIUS server and the trust RADIUS option is enabled, then the user session timeout is set to infinity; the user remains logged in indefinitely.

Note. Setting the session timeout value to zero is not configurable through the CLI; the value is only set to zero when zero is returned from the RADIUS server and the trust RADIUS option is enabled.

UNP Router Authentication User Group Guidelines

A router authentication user group specifies a destination network group and an optional source network group as a condition for identifying traffic that will be challenged for authentication. If a source group is specified, a destination group is required. However, if just a destination group is specified, a source group is not required.

- UNP router domain authentication becomes operational as soon as at least one router authentication user group is configured.
- User traffic destined for an IP network defined in the destination network group is challenged for authentication.
- A user can log in from multiple devices, but is challenged for authentication from each device if the device source IP is part of a source network group condition.
- Successful authentication would allow the user to access the IP network domain specified in the destination network group.
- A user is challenged for authentication for each destination network the user attempts to access, even if authentication was already granted or denied for other destination networks.

Captive Portal Profile Guidelines

Router domain authentication interacts with the internal Captive Portal mechanism supported on the OmniSwitch. An internal Web server on the local switch presents Captive Portal Web pages to obtain user credentials (see "Using Captive Portal Authentication" on page 30-91 for more information).

A Captive Portal profile is a configuration entity that provides flexible assignment of Captive Portal configuration parameters that are used for device authentication. If a Captive Portal profile is not assigned, then existing global authentication session parameter values are applied.

- The following authentication session parameters defined globally or in a Captive Portal profile are applied to Captive Portal authenticated traffic:
  - The RADIUS server to use for authentication.
  - Session timeout value (determines the aging of an accept/deny entry).
  - Captive Portal maximum retry attempts (number of login attempts the user is given before access is blocked).
  - Success redirect URL (the URL to display to the client after successful Captive Portal authentication).
- The following authentication session parameters defined globally or in a Captive Portal profile are applied to IP authenticated traffic:
  - The RADIUS server to use for authentication.
  - Session timeout value (determines the aging of an accept/deny entry).
- Only the authentication session parameters mentioned above are applied to router authentication user traffic; any other global or Captive Portal profile parameter values are ignored for this feature.

Configuring Router Authentication

The following steps provide a tutorial for configuring router domain authentication.

1 Use the **unp network-group** command to create a router authentication network group and assign IP network addresses to that group (do not include the internal Captive Portal IP subnet or the DNS server IP address in this group). For example:

```
-> unp network-group net-grp1 11.10.12.5 20.10.12.5 30.10.12.5
-> unp network-group net-grp2 40.5.5.1 50.5.5.1
```

2 Use the **unp router-auth user-group** command to create a router authentication user group network group condition. For example:

```
-> unp router-auth user-group ra-ugrp1 src-network-group net-grp1 dst-network-
group net-grp2
-> unp router-auth user-group ra-ugrp2 dst-network-group net-grp2
```

3 *Optional*. Use the **unp router-auth cp-profile** command to specify an existing Captive Portal profile to apply to router authentication users. For example:

```
-> unp router-auth cp-profile cp-1
```

If a Captive Portal profile is not specified, then the global authentication session parameter values are applied to router authentication users.

Configuring a Captive Portal Profile for Router Authentication

Configure a Captive Portal profile to define specific authentication session parameter values that are applied to user traffic that matches a user group network condition. Only the following authentication session parameters are applied to router authentication users; all other parameters are ignored:

- The RADIUS server to use for authentication.

- Session timeout value (determines the aging of an accept/deny entry).
- Captive Portal maximum retry attempts (number of login attempts the user is given before access is blocked).
- Success redirect URL (the URL to display to the client after successful Captive Portal authentication).

The following steps provide a brief tutorial for configuring a Captive Portal profile for router authentication:

1 The RADIUS server to use for router authentication is defined through the **aaa radius-server** and **aaa device-authentication** commands. For example:

```
-> aaa radius-server rad1 host 10.0.0.20 key automation
-> aaa device-authentication captive-portal rad1
```

2 The session timeout value defaults to 12 hours and is configurable through the **aaa session-timeout** command. For example:

```
-> aaa captive-portal session-timeout enable interval 13000
```

This configured value, however, is overridden if the RADIUS server returns a session timeout value of zero and the **trust-radius** option of the **aaa session-timeout** command is enabled. For example:

```
-> aaa captive-portal session-timeout enable interval 13000 trust-radius enable
```

3 Both the RADIUS server designation for Captive Portal and the session time out value can be assigned to an AAA profile through the **aaa profile** command. For example:

```
-> aaa profile ap-1 device authentication captive-portal rad1
-> aaa profile ap-1 captive-portal session-timeout 13000
```

Once created, the “ap-1” profile can then be assigned to a Captive Portal profile.

4 Use the **captive-portal-profile** command to configure a Captive Portal profile to apply to router authentication users. For example:

```
-> captive-portal-profile cp-p1 aaa-profile ap_1
-> captive-portal-profile cp-p1 retry-count 5
-> captive-portal-profile cp-p1 success-redirect-url http://server-1.com/
pass.html
```

If an AAA profile is not specified when creating a Captive Portal profile, then the global AAA parameter values are applied.

5 Use the **unp router-auth cp-profile** command to apply the “cp-p1” profile parameter values to router authentication users. For example:

```
-> unp router-auth cp-profile cp-1
```

If a Captive Portal profile is not specified, then the global values for these specific authentication session parameters are applied.

Verifying the Router Authentication Configuration

Use the **show unp network-group** command to display the UNP network group configuration. For example:

```
-> show unp network-group
  Network Group Name      IP-Address/Mask
-----------------------+-----------------------
net-grp1                10.0.0.1/255.255.0.0,
                        20.0.0.1/255.255.0.0,
                        30.0.0.1/255.0.0.0
net-grp2                40.0.0.1/255.255.0.0
net-grp3                10.0.0.1/255.255.0.0,
                        40.0.0.1/255.255.0.0

 Total Network-Group Count: 3
```

Use the **show unp router-auth user-group** command to display the UNP router authentication user group configuration. For example:

```
-> show unp router-auth user-group
User-Group Name                  Src-Network-Group          Dst-Network-Group
--------------------------+--------------------------+--------------------------
ra-ugrp1                    net-grp1                    net-grp2
ra-ugrp2                    net-grp3                    net-grp2
ra-ugrp3                    net-grp1                    net-grp3
ra-ugrp4                    net-grp2                    net-grp3
ra-ugrp5                                                net-grp4
```

Use the **show unp router-auth configuration** command to verify if a Captive Portal profile is configured to specify authentication session parameters for router authentication users. For example:

```
-> show unp router-auth configuration
       CP-Profile Name     : cp-prof1

       CP Params:
            Captive Portal AAA Profile Name        = a1
            Captive Portal Success Redirect URL    = http://server-1.com/pass.html
            Captive Portal Retry Count             = 3
```

Verifying and Managing Router Authentication Users

Use the **show unp router-auth users** command to display information about users authenticating through the router authentication process.

```
-> show unp router-auth users
   UserName   Destination    User-Group Intf-Name/  Auth Auth   LoginTime
              IP-Network                  Vlan      Type Status
-------------+--------------+----------+-----------+----+------+-------------------
Guest-user1   70.0.0.1       ra-ugrp1   L3-auth1:20  CP   Pass  06/20/2018 06:43:47
Employee-002  70.0.0.2       ra-ugrp2   L3-auth2/30  CP   Fail  06/50/2018 02:00:02
40.1.1.20     70.0.0.3       ra-ugrp3   L3-auth3/40  IP   Fail  07/20/2018 10:10:05
50.1.1.20     70.0.0.4       ra-ugrp4   L3-auth4/50  IP   Pass  07/10/2018 05:14:10

Total users : 4
```

Use the **unp router-auth user flush** command to clear the MAC addresses of the specified users. For example, use the **unp router-auth user flush** command with the **user-group** parameter to clear the MAC addresses of users that are associated with the "ra-ugrp1" user group:

```
-> unp router-auth user flush user-group ra-ugrp1
```

Router Domain Authentication Example

The use case described in this section provides an example of using role-based router domain authentication in a campus network where multiple users with different roles (teachers, administrative staff, students, network administrators, etc.) connect to a network. However, all users should not have access to all devices. For example, a student connecting to the network should not have access to security cameras. Using router domain authentication adds another layer of security to ensure that only authorized users can access specific devices, such as security cameras.

The following diagram shows an example of a campus network topology in which router domain authentication is configured to challenge users attempting to access devices within a specific IP network:

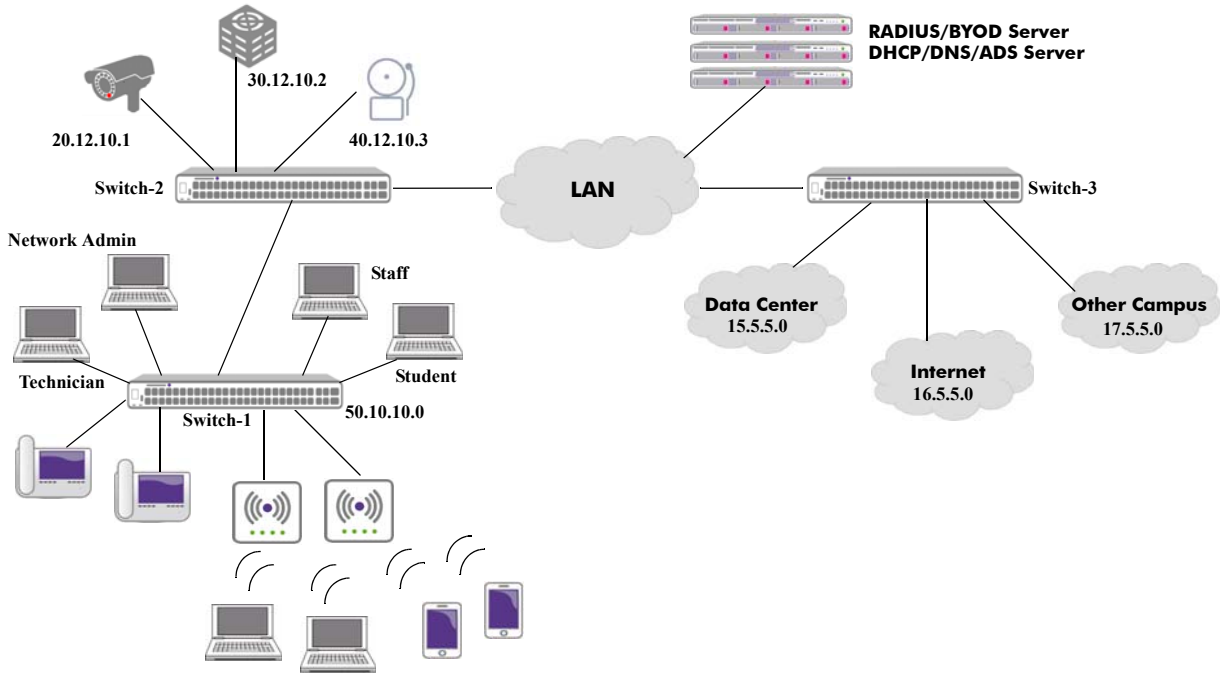


Figure 30-3 : Campus Network Topology Example

In this configuration example:

- Router domain authentication is configured on Switch-2 and Switch-3 to process users connected to Switch-1 that are attempting to access resources through Switch-2 and Switch-3.
- Users connected to Switch-1 are initially learned, authenticated, and classified into a UNP. The MAC address of each user is known to Switch-1. However, when a user sends traffic destined for resources accessible through Switch-2 and Switch-3, router authentication will challenge the user to authenticate again.
- When user traffic reaches Switch-2 or Switch-3, only the IP address of that user is known to Switch-2 or Switch-3; the source MAC address is not known. Router domain authentication is designed to authenticate traffic originating from one or more hops away based on the source IP address of the user and/or the IP network the user is attempting to access.

- OmniSwitch internal Captive Portal is configured on Switch-2 and Switch-3 for router authentication of users sending HTTP/HTTPS traffic. A Captive Portal profile is defined and assigned to the router authentication configuration.

The following commands provide an example of how role-based router domain authentication is configured on Switch-2 and Switch-3 as follows:

```
-> unp network-group secure-devices 20.12.10.1 30.12.10.2 40.12.10.3
-> unp network-group user-devices 50.10.10.0 mask 255.255.0.0

-> unp router-auth user-group students src-network-group user-devices dest-network-
group secure-devices

-> aaa radius-server rad1 host 10.0.0.20 key automation
-> aaa device-authentication captive-portal rad1
-> aaa captive-portal session-timeout enable interval 13000
-> aaa captive-portal session-timeout enable interval 13000 trust-radius enable
-> aaa profile ap-1 device authentication captive-portal rad1
-> aaa profile ap-1 captive-portal session-timeout 13000
-> captive-portal-profile cp-p1 aaa-profile ap_1
-> captive-portal-profile cp-p1 retry-count 5
-> captive-portal-profile cp-p1 success-redirect-url http://server-1.com/pass.html

-> unp router-auth cp-profile cp-1
```

Using Captive Portal Authentication

Captive Portal authentication is a mechanism by which user credentials are obtained through Web pages and authenticated through a RADIUS server. If the authentication is successful, the RADIUS server may return a role (policy list) that is applied to traffic from the user device. The OmniSwitch implementation supports an internal Captive Portal mechanism. An internal Web server on the local switch presents Captive Portal Web pages to obtain user credentials.

Internal Captive Portal authentication is a configurable option for a UNP profile that is applied after a user is assigned to the profile (after the initial 802.1X or MAC authentication or classification process). Captive Portal provides a secondary level of authentication that is used to apply a new role (QoS policy list) to the user.

- The RADIUS server returns the name of a QoS policy list or the name of a UNP profile that specifies a policy list name.
- If the RADIUS server does not return a QoS policy list or UNP profile name, a locally configured Captive Portal authentication pass policy specifies a QoS policy list or UNP profile name to assign to the user device.

The method for determining which QoS policy list is applied to a user device is based on the following precedence in descending order:

1 A policy list returned from the RADIUS server.

2 A domain specific policy list specified in the Captive Portal authentication pass configuration of a Captive Portal profile.

3 A policy list specified in the Captive Portal authentication pass configuration of a Captive Portal profile.

4 A domain specific policy list specified in the global Captive Portal authentication pass setting for the switch.

5 A policy list specified in the global Captive Portal authentication pass setting for the switch.

6 A policy list associated with a UNP profile returned from the RADIUS server.

7 A policy list associated with a domain specific UNP profile that is specified in the global Captive Portal authentication pass setting for the switch.

8 A policy list associated with a UNP profile that is specified in the global Captive Portal authentication pass setting for the switch.

An external, guest Captive Portal authentication mechanism is provided through the Access Guardian OmniSwitch integration with the Unified Policy Access Manager (UPAM) or the ClearPass Policy Manager (CPPM). See "Bring Your Own Devices (BYOD) Overview" on page 30-150 for more information.

This section provides the following information regarding configuring and using the OmniSwitch internal Captive Portal mechanism:

- "Configuration Tasks and Guidelines" on page 30-92
- "Quick Steps for Configuring Captive Portal Authentication" on page 30-93
- "Configuring the Captive Portal Operating Mode" on page 30-94

- "Using Captive Portal Configuration Profiles" on page 30-95
- "Replacing the Captive Portal Certificate" on page 30-96
- "Customizing Captive Portal Web Pages" on page 30-97
- "Authenticating with Captive Portal" on page 30-98.

Configuration Tasks and Guidelines

Consider the following tasks and guidelines when configuring the internal Captive Portal feature:

- **Define and map the RADIUS server to use for internal Captive Portal authentication**. The switch needs to know which RADIUS server to access to validate user credentials received through the Captive Portal Web pages. This is done through the authentication, authorization, and accounting (AAA) feature on the switch. See "Setting Authentication Parameters for the Switch" on page 30-34 for more information

- **Enable the AAA session timer for Captive Portal.** The session timer determines the amount of time a Captive Portal login session can remain active. *By default, this timer is disabled and must be enabled for Captive Portal sessions.* When enabled, the session timeout value defaults to 12 hours. To enable the session timer and change the timer value, if necessary, use the **aaa session-timeout** command.

- **Configure additional Captive Portal session parameters**. Default parameter values are in place to determine specific settings that apply to Captive Portal sessions, such as the number of login attempts allowed and an inactivity time limit. It is only necessary to change these global settings if the default values are not sufficient. See "Configuring Authentication Session Parameters" on page 30-36 for more information.

- **Configure the Captive Portal operating mode.** There are two operational modes supported: internal (the default) and internal with DHCP. When a device is in the pre-login state, the operating mode determines if DNS and DHCP traffic from that device is processed by an external server or processed locally on the switch. To change the operating mode, use the **captive-portal mode** command. See "Configuring the Captive Portal Operating Mode" on page 30-94 for more information.

- **Avoid using the Captive Portal IP subnet within the network.** This subnet is used exclusively by the Captive Portal feature to redirect DNS requests to the Captive Portal login screen. In addition, If Captive Portal is operating in the internal DHCP mode, then this subnet is also used to assign a temporary IP address for a client device that is attempting web-based authentication.

- **Change the Captive Portal IP subnet**. By default the Captive Portal subnet is set to 10.123.0.0 and the Captive Portal IP address is 10.123.0.1. If the internal DHCP mode is active, then addresses 10.123.0.2 through 10.123.0.254 are leased to DHCP clients. If a different Captive Portal subnet is required to avoid a conflict within the IP network, use the **captive-portal ip-address** command to change the IP subnet.

  Note. Make sure the DNS server configuration reflects the same Captive Portal redirect URL name and IP address that is configured for the OmniSwitch.

- **Configure a Captive Portal redirect URL**. The switch responds to initial HTTP/HTTPS requests from the user with a redirect URL. By default, this URL is set to "captive-portal.com". To change the redirect URL, use the **captive-portal name** command. To replace the default certificate with a well-known CA certificate, see "Replacing the Captive Portal Certificate" on page 30-96.

- **Configure a custom proxy port number for Captive Portal sessions.** Optionally, use the **captive-portal proxy-server-port** command to specify a proxy port number other than 8080 (the default).

- **Configure a UNP profile with Captive Portal authentication enabled.** The OmniSwitch Captive Portal process is triggered when a user device is classified into a profile on which Captive Portal authentication is enabled. For more information, see "Configuring UNP Profiles" on page 30-58.
- **Assign the QoS policy list to change the user role**. Captive Portal is a post-authentication and/or classification process that is used to dynamically change the user role (QoS policy list applied to the user). After the user successfully logs in, the RADIUS server returns a new policy list or UNP profile to apply to the user device. If the RADIUS server does not return a policy list or profile name, then the QoS policy list or profile name specified through the **captive-portal authentication-pass** command is used instead. This command can also be used to specify a domain-specific policy (the policy list or UNP profile is only applied to user devices from a specific domain).
- **Configure a redirect URL for successful Captive Portal login**. Optionally, use the **captive-portal success-redirect-url** command to redirect a user to a specific site after the user successfully logs in through the Captive Portal session. By default, no Captive Portal success redirect URL is configured.
- **Make sure that a standard browser is available on the client device**. No specialized client software is required. When a device is classified into a UNP profile that has Captive Portal enabled, the user device is placed into a pre-login role. In this role the user device is allowed to contact a DHCP server to obtain an IP address and contact a DNS server to resolve the Captive Portal URL to get the Captive Portal IP address, which is associated with the internal Web server on the switch. See "Authenticating with Captive Portal" on page 30-98 for more information.

Quick Steps for Configuring Captive Portal Authentication

The following procedure provides a brief tutorial for setting up the OmniSwitch implementation of Captive Portal authentication. For additional configuration tutorials, see the Captive Portal application examples on page 30-134 and page 30-136 in the "Access Guardian Application Examples" section.

1 Configure the RADIUS server to use for internal Captive Portal authentication. For example:

```
-> aaa radius-server cp-auth host 10.135.60.44 hash-key secret retransmit 3
timeout 2 auth-port 1812 acct-port 1813
-> aaa device-authentication captive-portal cp-auth
```

2 Configure the RADIUS server with the IP address of the OmniSwitch and the same shared secret that was assigned through the AAA RADIUS server configuration in Step 1.

3 Add the user name and password details in the RADIUS server.

4 Enable the Captive Portal session timer to determine the amount of time the user session remains active after a successful login (the default time is set to 12 hours). For example:

```
-> aaa captive-portal session-timeout enable
```

5 Configure a UNP profile and enable Captive Portal authentication on the profile. For example:

```
-> unp profile Captive-Portal
-> unp profile Captive-Portal captive-portal-authentication
-> unp profile Captive-Portal map vlan 111
```

6 Enable UNP functionality on the port that will connect the user device to the OmniSwitch and assign the profile created in Step 5 as the default profile for the port. For example:

```
-> unp port 1/1/20 port-type bridge
-> unp port 1/1/20 default-profile Captive-Portal
```

7 Change the Captive Portal redirect URL (defaults to captive-portal.com), if necessary. For example:

```
-> captive-portal name cp-cert
```

Note. Do not preface the redirect URL domain name with **https://**; the switch automatically adds **https://** to the beginning of the domain name.

8 Change the Captive Portal IP address (defaults to 10.123.0.1), if necessary. For example:

```
-> captive-portal ip-address 10.255.0.20
```

9 Select the internal Captive Portal operating mode (the default) or the internal Captive Portal with DHCP operating mode. For example:

```
-> captive-portal mode internal
-> captive-portal mode internal-dhcp
```

10 Configure the DNS servers with a mapping between "captive-portal.com" and the10.123.0.1 IP address. The user device resolves this URL through access to the DNS server to get the Captive Portal IP address, which is mapped to the internal Web server on the OmniSwitch.

Verifying the Captive Portal Setup

1 Make sure the client has received an IP address from the DHCP server.

2 When the client opens a Web browser and attempts to access any URL, the client is prompted with the Captive Portal login page. This is the Web page presented by the internal server on the OmniSwitch.

3 When the client enters the appropriate login credentials and clicks on the "Submit" button on the login page, the client is presented with the Captive Portal status page. This page indicates that the login was successful and the remaining session time.

4 Use the **show unp user status** command on the OmniSwitch to display the status of the Access Guardian classification and Captive Portal authentication process for the MAC address of the client.

Configuring the Captive Portal Operating Mode

There are two Captive Portal operating modes supported: internal and internal DHCP. An important difference between these two modes is how a VLAN change is handled when successful Captive Portal authentication results in a new VLAN assignment for the device.

- When the internal Captive Portal mode is active (the default), a port bounce action is required for the device to obtain an IP address within the new VLAN domain.
  - A port bounce action only applies to a MAC authenticated non-supplicant (non-802.1X device). If the device is a supplicant (802.1X device), then an EAP-Fail frame is sent instead. In both cases, re-authentication is triggered for both types of devices.
  - Existing BYOD global commands are leveraged to configure the port bounce action.
- When the internal DHCP Captive Portal mode is active, a port bounce action is not required for the device to obtain an IP address within the new VLAN domain. This is especially useful for ports on which multiple users are authenticated.

A user device is placed in a pre-login state when the device is assigned to a UNP profile that has Captive Portal authentication enabled. In this state the device gets an IP address and the DNS server address.

- If the internal Captive Portal mode is active, the device contacts a DHCP server directly to get an IP address and contacts the DNS server for resolution of the redirect name to get the configured Captive Portal IP address. If a subsequent VLAN change occurs, a port bounce action is triggered to initiate a new request for an IP address. If multiple user devices are authenticating through the same port, all of the devices are disrupted.
- If the internal DHCP Captive Portal mode is active, the switch provides basic DHCP functionality to assign a short-term leased IP address to the device from the configured Captive Portal subnet and provide the necessary DNS information. A short-term lease for the IP address ensures that the lease will expire soon and the device can initiate a request for a new IP address from within the new VLAN domain. This avoids the need for a port bounce action to get a new IP address when there is a VLAN change.

To change the Captive Portal operating mode, use the **captive-portal mode** command. For example:

```
-> captive-portal mode internal-dhcp
```

Configurable parameters for the internal DHCP Captive Portal mode include lease time, renew time, and rebinding time. By default, these parameter values are set to 30 seconds, 15 seconds, and 26 seconds, respectively. To change these values, use the **captive-portal mode** command with the optional **ip-lease-time**, **ip-renew-time**, and **ip-rebinding-time** parameters. For example:

```
-> captive-portal mode internal-dhcp ip-lease-time 120
```

Consider the following when changing the internal DHCP parameter values:

- The IP renew time is 50% of the IP lease time.
- The IP rebinding time is 87.5% of the IP lease time.
- When only the IP lease time is changed, the IP renew time and IP rebinding time are automatically recalculated based on the noted percentages.
- Make sure the IP renew time specified is less than the IP rebinding time.
- Make sure the IP rebinding time specified falls between the IP renew time and IP lease time.

Use the **show captive-portal configuration** command to display the Captive Portal mode configuration.

For more information about the commands described in this section, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Using Captive Portal Configuration Profiles

A Captive Portal profile is a configuration entity that provides flexible assignment of Captive Portal configuration parameters to devices classified into specific UNP profiles. However, this type of profile is only valid when assigned to profiles on which Captive Portal authentication is enabled.

When a Captive Portal profile is applied to a UNP profile, the parameter values defined in the Captive Portal profile override the global Captive Portal parameter values configured for the switch. If there is no Captive Portal profile associated with a profile, then the global Captive Portal configuration is applied.

Use a Captive Portal profile to define and apply the following Captive Portal configuration settings for user sessions associated with a specific UNP profile:

- The name of an AAA profile to define specific device authentication configuration options, such as which servers to use for Captive Portal authentication and parameter values for session timers and

RADIUS attributes. If there is no AAA profile assigned, the global AAA configuration for the switch is used.

- A URL to which user devices are redirected when Captive Portal authentication is successful.
- The number of login attempts allowed for the Captive Portal session.
- The name of the QoS policy list or UNP profile to apply when Captive Portal authentication is successful but the RADIUS server did not return a policy list or profile name.

Captive Portal profiles can be used to apply a custom Captive Portal configuration to different sets of user devices based on the UNP profile assignment for the device.

Configuring Captive Portal Profiles

Use the **captive-portal-profile** command to create a profile name and configure parameter values for that profile. For example:

```
-> captive-portal-profile cp_p1
-> captive-portal-profile cp_p1 aaa-profile aaa_p1
-> captive-portal-profile cp_p1 authentication-pass realm prefix domain asia-
pacific policy-list list1

-> captive-portal-profile cp_p2 retry-count 5
-> captive-portal-profile cp_p2 authentication-pass profile cp-pass
-> captive-portal-profile cp_p2 authentication-pass profile-change enable
-> captive-portal-profile cp_p2 success-redirect-url http://server-1.com/
pass.html
```

Captive Portal profiles are only valid for UNP profiles on which Captive Portal authentication is enabled. Use the **unp profile captive-portal-authentication** command to enable Captive Portal authentication for a UNP profile. For example:

```
-> unp profile cp_unp captive-portal-authentication
```

Use the **unp profile captive-portal-profile** command to assign a Captive Portal configuration profile to a UNP profile. For example:

```
-> unp profile cp_unp captive-portal-profile cp_p1
```

Use the **show captive-portal profile-names** command to display the Captive Portal profile configuration.

For more information about the commands described in this section, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Replacing the Captive Portal Certificate

By default, the OmniSwitch uses a built-in, self-signed certificate for Captive Portal. The certificate is named “default_cportalCert.pem” and is stored in the “/flash/switch” directory on the switch. To replace the default certificate with a well known CA certificate, use the following steps:

1. Backup the existing default certificate.

```
-> cp default_cportalCert.pem default_cportalCert.pem.old
```

2. Rename the new well known CA certificate file to “default_cportalCert.pem”.
3. Copy the certificate file to the “/flash/switch” directory.

4 Use the **captive-portal name** command to reload the Web configuration (use the CN name as specified in the new certificate):

```
-> captive-portal name CN_name
```

5 Attempt a captive portal log in to verify the change.

Note. The certificate must be in the x509 format. To generate an x509 formatted certificate (.pem), perform the following on a Linux or Unix machine:

1 Have the private key and the CA signed certificate available.
2 Issue the "cat *privateKey ca_certificate* | tee *switch_cert_file*"(i.e default_cportalCert.pem) command.

Customizing Captive Portal Web Pages

The Web pages that Captive Portal presents to users are stored in the "/flash/switch/captive_portal/release_files/" directory on the switch. To present the user with customized Web pages:

1 Create a folder in the same path as the "release_files" folder and name the new folder "custom_files" (for example "/flash/switch/captive_portal/custom_files/").

2 Copy the "assets" and "templates" folders found under "/flash/switch/captive_portal/release_files/" to the "custom_files" folder.

3 Modify the contents in the copied folders to create custom Web pages.

4 Once the custom files are created with the images and information the file type requires, download the files to the "/flash/switch/captive_portal/custom_files" folder.

5 Enable Captive Portal customization using the **captive-portal customization** command.

```
-> captive-portal customization enable
```

When a Captive Portal session is initiated, the switch checks to see if there is a "custom_files" folder; if so, then the files in that folder are incorporated presented to the user. If there are no files found or the "custom_files" folder does not exit, the default Web page components found in the "release_files folder are incorporated and presented to the user.

Consider the following guidelines when customizing Captive Portal Web page components:

- The "release_files" folder is overwritten each time the switch reboots; ***DO NOT*** modify the files in this folder for custom use.
- The folders "assets" and "templates" under the "/flash/switch/captive_portal/custom_files/" directory are used to create and display Web pages to Captive Portal users when the switch reboots or at runtime when Captive Portal customization is enabled for the switch, if the "custom_files" folder exists.
- Anything in the custom "assets" folder is statically served by the internal Web server on the switch whenever they are requested. These pages are typically **.css** files, javascript files, or the acceptable use policy and are linked to files in the custom "templates" folder.
- The custom "templates" folder contains the Web pages that are dynamically served to users depending on the Captive Portal state of each user. The file names in this folder must not be changed. The login form field names and form action in these pages must not be changed. The variables in these pages, as denoted by "<?=$(name)?>", are substituted in place by the internal Web server.
- Filenames are case sensitive. When creating a custom file, ensure that the filename matches the filename exactly as shown in the "release_files" folder.

- Enabling Captive Portal customization automatically triggers the use of the custom Web pages at runtime. There is no need to reboot the switch to trigger this action.

The following is an example of a customized Captive Portal login page:

Figure 30-4 : Customized Captive Portal login page

Authenticating with Captive Portal

Access Guardian determines that a client device is a candidate for Web-based authentication if the following conditions are true:

- The device is connected to a UNP-enabled port.
- The device is assigned to a UNP profile on which Captive Portal authentication is enabled.

When these authentication conditions are met, Access Guardian places the device MAC address into a Captive Portal pre-login state. In this state, the device is allowed to directly contact a DHCP server to get an IP address and get the DNS server address.

Next, the user opens a Web browser and the initial HTTP/HTTPS requests are responded to with the Captive Portal redirect name. The user device contacts the DNS server to resolve the redirect name and receives the configured Captive Portal IP address. Requests are then sent to the Captive Portal IP address that is mapped to the internal OmniSwitch Web server. The internal server responds to the HTTP/HTTPS requests by presenting a Captive Portal login page to the user device.

Logging Into the Network with Captive Portal

Once a user device is in the Captive Portal state, the following steps are required to complete the authentication process:

1 Open a Web browser window on the client device. If there is a default home page, the browser attempts to connect to that URL. If a default home page is not available, enter a URL for any website and attempt to connect to that site.

A certificate warning message may appear when the Web browser window opens. If so, select the option to continue on to the website.

When the browser window opens and after the certificate warning message, if any, is cleared, Captive Portal displays a login screen similar to the one shown in the following example:

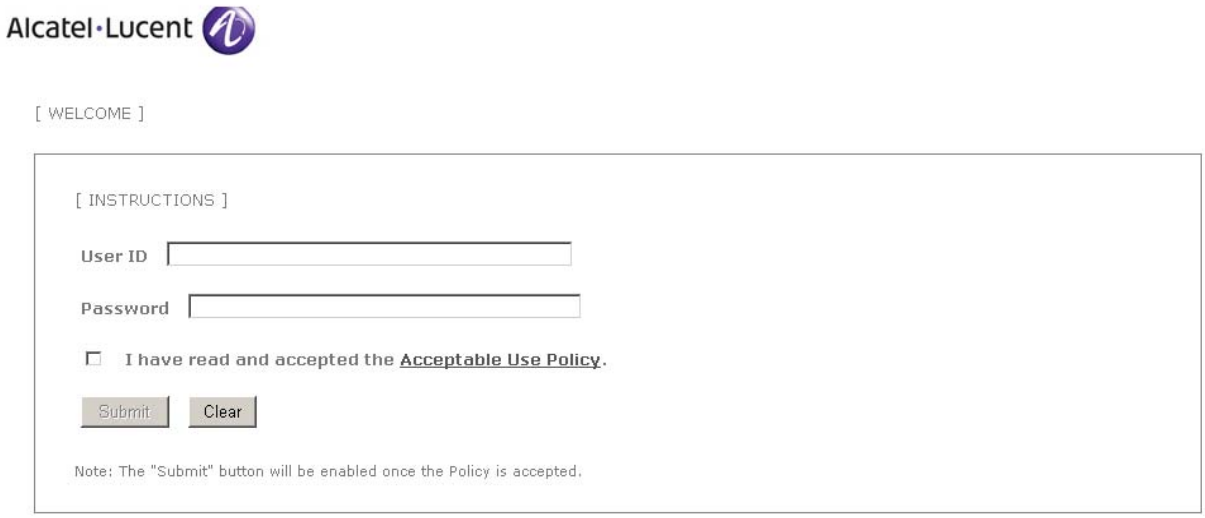


Figure 30-5 : Captive Portal - Login Screen

2 Enter the user name in the “User ID” field.

3 Enter the user password in the “Password” field.

4 Click on the “Acceptable Use Policy” box to activate the “Submit” button.

5 Click the “Submit” button to login to the network. When the “Submit” button is clicked, Captive Portal sends the user information provided in the login window to the RADIUS server for authentication.

6 If user authentication is successful, the following status and logout messages are displayed:

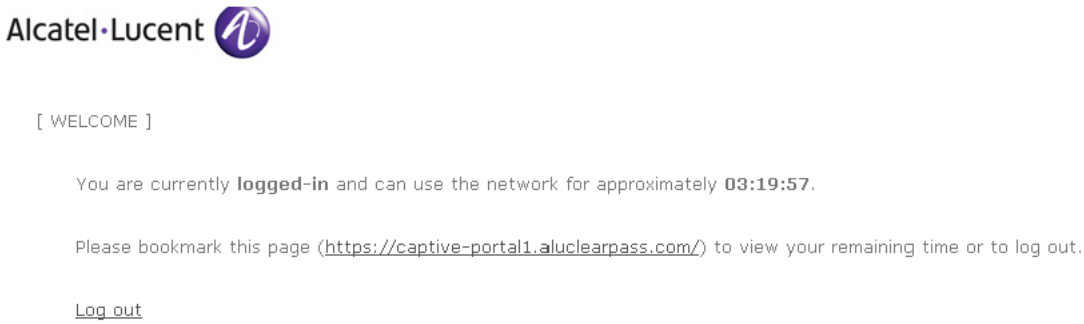


Figure 30-6 : Captive Portal - Authentication is successful

The user is now logged into the network and has access to all network resources as determined by the Captive Portal role (QoS policy list) assigned to the user. The original profile and associated VLAN membership for the user was not changed; only the QoS policy list returned from the RADIUS server is applied to the user.

7 Before leaving the Captive Portal status page (shown in Step 6) or closing the browser window, make note of the URL presented on the status page.

Logging Off the Network with Captive Portal

Click on the "Log out" button on the Captive Portal login status page. If this page is not displayed, go to the bookmarked URL provided in Step 6 of the login procedure on page 30-99. When the URL is entered in the location bar of the browser or the URL bookmark is selected, the Captive Portal login status page is displayed.

When the "Log out" button is selected, the user is logged off the network and the user device returns to a Captive Portal pre-login state. The user is then presented with the Captive Portal login page.

Note. A user is automatically logged out of the network if the Captive Portal session time limit is reached. For more information, see "Configuring Authentication Session Parameters" on page 30-36.

OmniAccess Stellar AP Integration

Access Guardian provides the framework through which OmniAccess Stellar Access Points (APs) connected to an OmniSwitch are detected, learned, and managed. Wireless client traffic is then forwarded from the AP to the OmniSwitch and onto the wired network. This integration provides a unified wireless over wired network access solution.

This section contains the following information about the OmniSwitch feature components that are required to discover and integrate wireless traffic that is forwarded by OmniAccess Stellar AP devices into an OmniSwitch network:

- "How it Works" on page 30-26.
- "AP Mode Configuration Guidelines - VLAN Domain" on page 30-104.
- "AP Mode Configuration Guidelines - SPB Service Domain" on page 30-106.
- "OmniAccess Stellar AP Configuration Guidelines" on page 30-107.
- "Quick Steps for Configuring OmniSwitch AP Discovery" on page 30-108.
- "Verify the OmniSwitch Configuration" on page 30-110.

How it Works

The OmniSwitch boots up with specific default configuration and operational settings that trigger the following process to detect, learn, and classify connected Stellar AP devices:

1 The switch and any Stellar AP device that is connected to a UNP bridge or access port initially exchange Link Layer Detection Protocol (LLDP) TLV packets. Through this exchange of LLDP packets, the switch identifies and learns the device MAC address as an AP.

- On UNP bridge ports, the AP device and AP clients are learned into the VLAN domain.
- On UNP access ports, the AP device and AP clients are learned into the Shortest Path Bridging (SPB) service domain.

2 If the AP mode is enabled for the UNP port and an AP device is detected on that port, the following actions are triggered to automatically change the operational status of the specified options (the operational status overrides the configured status):

- The transmission of LLDP Port VLAN ID and AP Location TLVs is operationally enabled on the UNP bridge or access port.
- The trust tag status for the UNP bridge port is operationally enabled. The trust tag status does not apply to UNP access ports.
- The global status for dynamic VLAN configuration is operationally enabled for the switch. Dynamic VLAN configuration does not apply in the SPB service domain.

3 Once the AP MAC address is detected and learned, a built-in LLDP UNP classification rule for access points classifies the AP device into one of the following built-in default profiles:

- **defaultWLANProfile** for AP devices connected to UNP bridge ports. This profile is mapped to a VLAN into which the AP device is classified. A VLAN-port association (VPA) is established between the UNP bridge port and profile VLAN on which the AP MAC address is learned and forwarded.
- **defaultWLANAccessProfile** for AP devices connected to UNP access ports. The profile is mapped to SPB service parameters. When the AP device is classified into this profile, an SPB Service Access Point (SAP) is dynamically created. A SAP-port association is established between the UNP access port and the SAP on which the AP MAC address is learned and forwarded.

4 After the AP device connection is established, classified, and the management VLAN or SPB service is assigned, any of the following actions can occur:

- The AP device sends DHCP packets.
- The switch transmits LLDP packets to the AP device to advertise the management VLAN or SPB service and AP location information.
- On a UNP bridge port, the AP device starts to send client-tagged traffic (tagged with the SSID VLAN). The switch will trust the VLAN tag of the AP client traffic and attempt to assign the traffic to a switch VLAN that matches the tag of the client traffic. If a matching switch VLAN does not exist, then the switch will dynamically create the necessary VLAN on which to forward the AP client traffic.
- On a UNP access port, the AP device starts to send client-tagged traffic (tagged with the SSID VLAN). The switch will attempt to assign the AP client traffic to a UNP service profile configured with an SPB service tag value that matches the VLAN tag of the client traffic. If a matching UNP service profile does not exist, then the AP client traffic is assigned to the System Default profile that will dynamically create the SPB service SAP on which to forward the AP client traffic.

5 MVRP will then propagate the VLAN configuration (AP management VLAN and any static or dynamic VLAN that was automatically tagged to carry AP client traffic) to adjoining switches in the network. This process creates specific VLAN domains through which the untagged AP management traffic and tagged wireless client traffic is forwarded on the wired network. This process is not triggered for the SPB service domain.

The OmniSwitch detection and integration of OmniAccess Stellar APs results in a switch configuration that includes a management VLAN or SPB service for the AP device and additional VLANs or SPB services for wireless client-tagged traffic that is forwarded by the AP onto the wired network.

The following diagram shows an example of a network topology in which a Stellar AP connected to an OmniSwitch UNP bridge port serves as a bridge between a wireless and wired network:

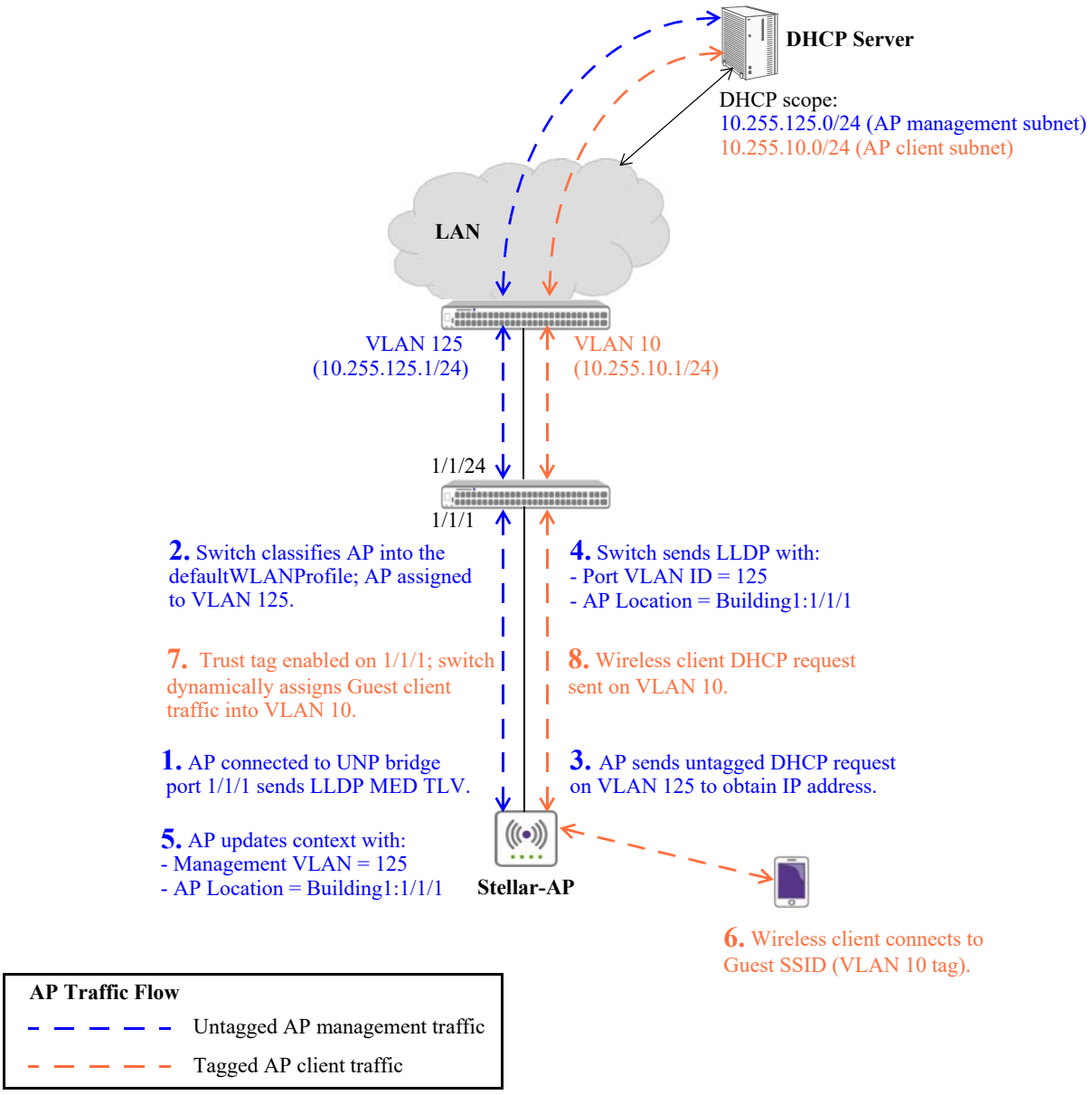


Figure 30-7 : OmniSwitch AP Discovery and Integration Example - VLAN Domain

The following diagram shows an example of a network topology in which a Stellar AP connected to an OmniSwitch UNP access port serves as a bridge between a wireless and wired network:

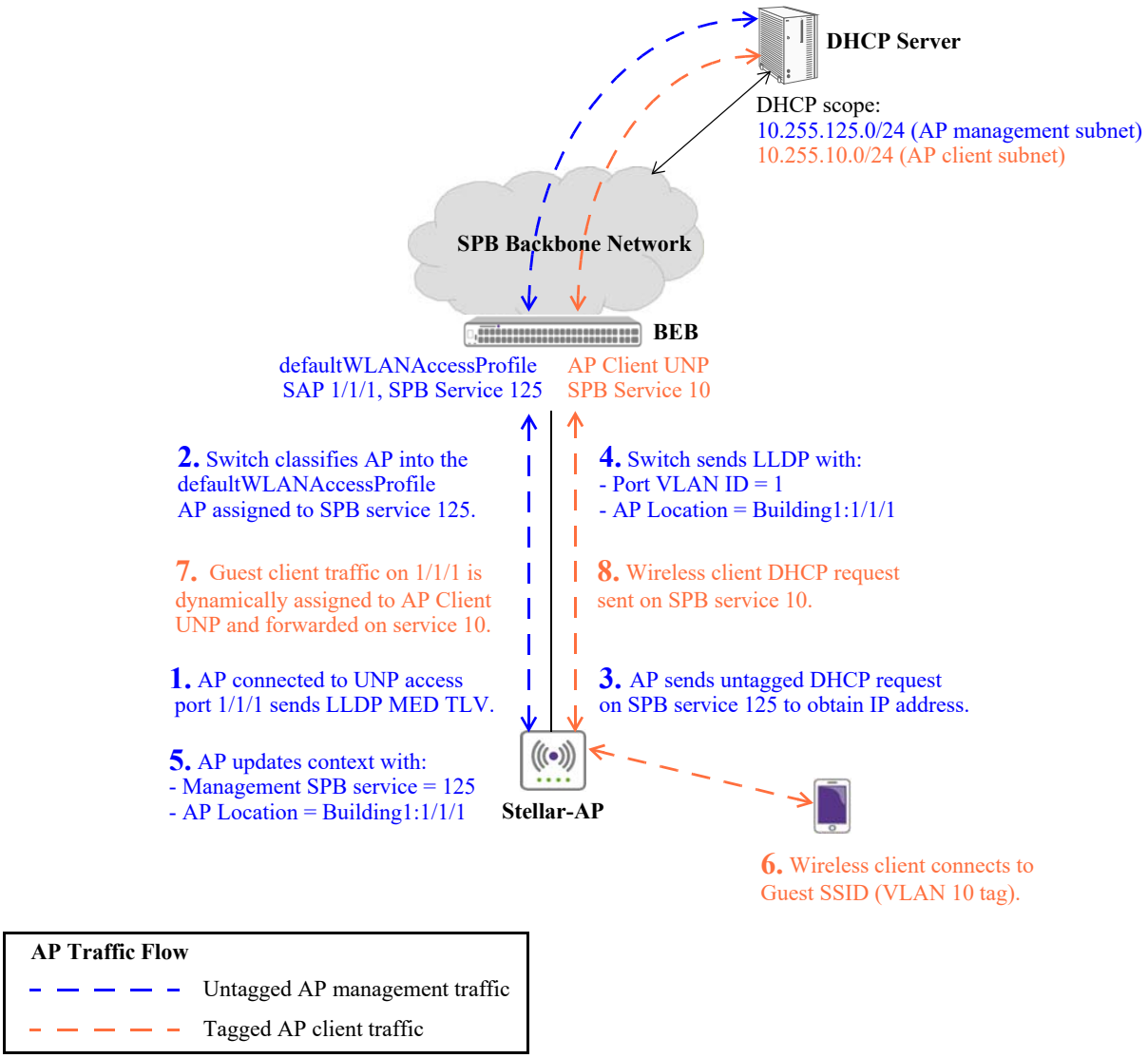


Figure 30-8 : OmniSwitch AP Discovery and Integration Example - SPB Domain

AP Mode Configuration Guidelines - VLAN Domain

The following information applies when an OmniAccess Stellar AP device is connected to a UNP bridge port. The AP device and AP clients are learned into the VLAN domain.

- **Link Layer Detection Protocol (LLDP) parameters**. The first packet a connected AP device sends should be an LLDP-MED TLV that identifies the device as an AP. When the AP device is detected on the UNP port, the switch sends LLDP packets to the AP device to communicate the management VLAN (LLDP Port Vlan ID TLV) and the AP Location (LLDP Proprietary TLV).
  - The management VLAN advertised to the AP device is the VLAN associated with the UNP profile to which the AP device is classified.
  - The AP Location advertised to the AP device is derived from local switch information (such as the UNP port, chassis MAC address, system name, system location).

- **UNP port parameters.** The port to which an AP device connects must be configured as a UNP bridge port with the AP mode enabled. The trust tag option for the UNP port is operationally enabled so that any tagged traffic coming from the AP device will automatically be trusted. This ensures that client-tagged traffic sent from the AP is forwarded on the VLAN domain that corresponds with the VLAN tag of the wireless client traffic.
  - A tagged MAC address is classified into the matching tagged VLAN. If that VLAN does not exist on the switch, a dynamic VLAN is created. For example, if the customer tag is VLAN 200 but this VLAN does not exist, the switch will dynamically create VLAN 200 to accommodate the client-tagged traffic.
  - When an AP MAC address is detected on a UNP port, the switch will flush all other MAC addresses previously learned on that same port. This ensures that the AP MAC address is always the first MAC address learned on that port; a requirement that designates the UNP port as an AP detected port.
  - By default, 802.1X and MAC authentication are enabled on UNP ports. If authentication of an AP device is not required, disable one or both of these options.
  - The default AP mode status for a port is derived from the global AP mode status at the time the port is configured as a UNP bridge port. If the global status is disabled, the port-level status defaults to disabled; if the global status is enabled, the port-level status defaults to enabled. The AP mode status can then be enabled or disabled for a specific UNP bridge port.
- **WLAN access role profile (defaultWLANProfile).** The “defaultWLANProfile” is a built-in profile that is designated for classifying Stellar AP devices. This profile is automatically assigned to a built-in UNP LLDP classification rule for APs that will recognize active AP devices connected to the switch and assign them to the “defaultWLANProfile”. The VLAN that is mapped to this profile will serve as the management VLAN for the classified AP devices.
  - The LLDP UNP classification rule for access points and the “defaultWLANProfile” are both *implicitly* configured on the switch. However, mapping a VLAN to the “defaultWLANProfile” requires *explicit* configuration.
  - Using the “defaultWLANProfile” to classify AP devices ensures that all of the AP devices connected to each switch in the wired network will use the same management VLAN.
  - The “defaultWLANProfile” is similar to a standard UNP VLAN profile except that the profile cannot be deleted; it is a built-in profile that is always available in the switch configuration.
  - The “defaultWLANProfile” does not appear in the configuration snapshot for the switch. However, when the default value for any of the configurable profile attributes is modified, then the profile settings will appear in the configuration snapshot.
- **WLAN access role profile (defaultWLANProfile) attributes.** In addition to the VLAN mapping, only the following profile attributes are configurable for the “defaultWLANProfile”:
  - **QoS policy list**. By default, there is no policy list assigned to a profile. Optionally assign a QoS policy list to apply further network access control to an AP device.
  - **Authentication flag**. By default, the Layer 2 authentication flag for a profile is disabled. Optionally enable the authentication flag to specify that only Layer 2 (802.1X or MAC) authenticated AP devices are allowed into the profile.
  - **Mobile tag**. By default the mobile tag status is disabled for a profile. Optionally enable the mobile tag status to specify that the first user that is learned on a UNP port and classified into the specified UNP profile will cause the UNP port to be added as a tagged member of the VLAN associated with the profile.
- **Dynamic VLAN configuration**. The switch operationally enables dynamic VLAN configuration to ensure that when the VLAN tag of AP client-tagged traffic does not match an existing switch VLAN, the switch will dynamically create the VLAN.
  - Dynamic VLANs created by UNP are identified as a separate type of VLAN; the default name is set to “UNP-DYN-VLAN” and the designated type is set to “UNPD”.

- The switch will automatically remove dynamically created VLANs when all the MAC addresses learned on the VLAN have aged out.

- **Multiple VLAN Registration Protocol (MVRP).** By default, MVRP is disabled for the switch. Enable this protocol to ensure the propagation of the AP management VLAN and any VLAN (static or dynamic) that was automatically tagged to carry AP client traffic to other switches.
  - AP management VLANs dynamically created by MVRP are converted to UNP dynamic VLANs (type UNPD).
  - If an AP client tag matches a VLAN that was dynamically created by MVRP, then that VLAN is converted to a UNP dynamic VLAN (type UNPD).

AP Mode Configuration Guidelines - SPB Service Domain

The following information applies when an OmniAccess Stellar AP device is connected to a UNP access port. The AP device and AP clients are learned into the Shortest Path Bridging (SPB) service domain.

- **Link Layer Detection Protocol (LLDP) parameters**. The first packet a connected AP device sends should be an LLDP-MED TLV that identifies the device as an AP. The first packet received on the UNP access port should be the LLDP-MED TLV from the AP device.
  - When the AP device is detected on the UNP port, the switch sends LLDP packets to the AP device to communicate the management SPB service (LLDP Port Vlan ID TLV) and the AP Location (LLDP Proprietary TLV).
  - The management SPB service advertised to the AP device is the service associated with the UNP profile to which the AP device is classified.
  - The AP Location advertised to the AP device is derived from local switch information (such as the UNP port, chassis MAC address, system name, system location).

- **UNP port parameters.** The port to which an AP device connects must be configured as a UNP access port with the AP mode enabled.
  - Assign a Layer 2 profile to the UNP access port that has the action for LLDP (80.1AB) control frames set to “peer”.
  - When an AP MAC address is detected on a UNP port, the switch will flush all other MAC addresses previously learned on that same port. This ensures that the AP MAC address is always the first MAC address learned on that port; a requirement that designates the UNP port as an AP detected port.
  - By default, 802.1X and MAC authentication are enabled on UNP ports. If authentication of an AP device is not required, disable one or both of these options.
  - Any MAC or 802.1X authentication that is enabled on the UNP port is bypassed for learning AP clients. However, the other Layer 2 options for learning can be configured and applied to AP client MAC addresses (such as enabling rule classification, assigning a default SPB service profile, or dynamic SPB System Default profile configuration).
  - The default AP mode status for a port is derived from the global AP mode status at the time the port is configured as a UNP access port. If the global status is disabled, the port-level status defaults to disabled; if the global status is enabled, the port-level status defaults to enabled. The AP mode status can then be enabled or disabled for a specific UNP bridge port.

- **WLAN access role profile (defaultWLANAccessProfile).** The “defaultWLANAccessProfile” is a built-in profile that is designated for classifying Stellar AP devices. This profile is automatically assigned to a built-in UNP LLDP classification rule for APs that will recognize active AP devices connected to the switch and assign them to the “defaultWLANAccessProfile”. The SPB service that is mapped to this profile will serve as the management service for the classified AP devices.
  - The LLDP UNP classification rule for access points and the “defaultWLANAccessProfile” are both *implicitly* configured on the switch. However, mapping SPB service parameters to the “defaultWLANAccessProfile” requires *explicit* configuration.

- SPB service parameter values that are mapped to “defaultWLANAccessProfile” cannot be removed. However, existing parameter values can be overwritten by configuring new profile values.
- Using the “defaultWLANAccessProfile” to classify AP devices ensures that all of the AP devices connected to each switch in the wired network will use the same management service.
- The “defaultWLANAccessProfile” is similar to a standard UNP service profile except that the profile cannot be deleted; it is a built-in profile that is always available in the switch configuration.
- The “defaultWLANAccessProfile” does not appear in the configuration snapshot for the switch. However, when the default value for any of the configurable profile attributes is modified, then the profile settings will appear in the configuration snapshot.

- **WLAN access role profile (defaultWLANAccessProfile) attributes.** In addition to the SPB service mapping, only the following profile attributes are configurable for the “defaultWLANAccessProfile”:
  - **QoS policy list**. By default, there is no policy list assigned to a profile. Optionally assign a QoS policy list to apply further network access control to an AP device.
  - **Authentication flag**. By default, the Layer 2 authentication flag for a profile is disabled. Optionally enable the authentication flag to specify that only Layer 2 (802.1X or MAC) authenticated AP devices are allowed into the profile.
  - **Mobile tag**. By default the mobile tag status is disabled for a profile. Optionally enable the mobile tag status to specify that the first user that is learned on a UNP port and classified into the specified UNP profile will cause the UNP port to be added as a tagged member of the service associated with the profile.
- **UNP Service Profiles for AP Clients**. There are no built-in service profiles or classification rules for AP clients, so UNP profiles mapped to SPB service parameters and classification rules to assign the AP clients to these profiles must be created.
  - Create SPB service profiles each with a VLAN tag to match the VLAN tag of the AP client frames.
  - Create a corresponding VLAN tag classification rule to assign the tagged AP client traffic to the appropriate service profile.
  - If there are no classification rules to capture the AP client traffic, the AP client traffic is assigned to the dynamically created SPB System Default profile associated with the UNP access port.

OmniAccess Stellar AP Configuration Guidelines

The Stellar AP device must meet the following configuration and operational requirements to ensure successful discovery and integration into an OmniSwitch network:

- The AP device must connect to an OmniSwitch UNP bridge or access port on which the AP mode status is enabled. This triggers the Access Guardian process to detect and integrate the AP device.
- The first packet a connected AP device sends (before an 802.1X or DHCP packet) must be an LLDP frame with the “WLAN AP” bit set in the System Capabilities TLV and the LLDP media TLV device type set as 4("Endpoint class 1V"). If this requirement is not met, the device may not get properly identified as an AP; this could trigger a different process for classifying the device MAC address or cause the address to be filtered.
- The AP device should always get an IP address using DHCP. Configure the DHCP server to use Option 138 for the scope of IP addresses that will be assigned to AP devices.
- AP management traffic is always sent untagged on the management VLAN or SPB service advertised by the switch.
- The wireless client traffic forwarded by the AP is always tagged with the assigned SSID VLAN. Access Guardian will automatically assign the client traffic to switch VLAN or SPB service profiles that correspond to the VLAN tags of the client traffic.

Quick Steps for Configuring OmniSwitch AP Discovery

The following procedures provide a brief tutorial for configuring existing OmniSwitch features to discover and interact with OmniAccess Stellar APs.

- "Quick Steps for Configuring AP Discovery in the VLAN Domain" on page 30-108.
- "Quick Steps for Configuring AP Discovery in the Service Domain" on page 30-109.

Quick Steps for Configuring AP Discovery in the VLAN Domain

1 Create the VLAN that will serve as the AP management VLAN on each participating switch in the network. For example:

```
-> vlan 200 name "AP Management VLAN"
```

2 Tag switch ports that connect to other switches with the VLAN created in Step 1. For example,

```
-> vlan 200 members port 1/1/24 tagged
```

3 Map the VLAN created in Step 1 to the built-in "defaultWLANProfile". For example:

```
-> unp profile defaultWLANProfile map vlan 200
```

4 Configure any switch port that will connect to a Stellar AP device as a UNP bridge port. For example:

```
-> unp port 1/1/12 port-type bridge
```

5 If necessary, enable the UNP AP mode for the UNP bridge port. For example:

```
-> unp port 1/1/12 ap-mode
```

If the global AP mode status was enabled at the time port 1/1/12 was configured as a UNP bridge port (Step 4), then the port-level status is already enabled and Step 5 can be skipped. Use the **show unp global configuration** command to verify the global AP status.

6 Enable MVRP for the switch to facilitate the propagation of the AP management VLAN and AP client VLANs. For example:

```
-> mvrp enable
```

7 Optionally disable authentication on the UNP port if authentication of the AP device is not required. For example:

```
-> no unp port 1/1/12 802.1x-authentication
-> no unp port 1/1/12 mac-authentication
```

8 Optionally configure the QoS policy list, authentication flag status, or mobile tag status for the "defaultWLANProfile" (changing other parameter values for this profile is not supported). For example:

```
-> unp profile defaultWLANProfile qos-policy-list qlist1
-> unp profile defaultWLANProfile authentication-flag

-> unp profile defaultWLANProfile inactivity-interval 200
ERROR: defaultWLANProfile parameters cannot be modified. Mobile-Tag, Qos-policy-
list, Authentication are only allowed
```

9 Optionally configure the system name, system location, and port alias. The information from one or more of these settings is used to derive the AP Location information that is transmitted by the switch to the connected AP device.

```
-> system name BWIAPS01
-> system location BWI Airport Hotel
-> interfaces port 1/1/12 alias BWI-AP01
```

Quick Steps for Configuring AP Discovery in the Service Domain

1 Configure a Layer 2 profile with a “peer” action defined for 802.1AB control frames. For example:

```
-> service l2profile “ap-SvcUnp” 802.1ab peer
```

2 Configure any switch port that will connect to a Stellar AP device as a UNP access port and assign the Layer 2 profile created in Step 1 to that port. For example:

```
-> unp port 1/1/15 port-type access
-> unp port 1/1/15 l2-profile ap-SvcUnp
```

3 If necessary, enable the UNP AP mode for the UNP access port. For example:

```
-> unp port 1/1/15 ap-mode
```

If the global AP mode status was enabled at the time port 1/1/15 was configured as a UNP access port (Step 2), then the port-level status is already enabled and Step 3 can be skipped. Use the **show unp global configuration** command to verify the global AP status.

4 Optionally disable authentication on the UNP port if authentication of the AP device is not required (authentication is enabled by default). For example:

```
-> no unp port 1/1/15 802.1x-authentication
-> no unp port 1/1/15 mac-authentication
```

5 Map SPB service parameters to the built-in “defaultWLANAccessProfile”. For example:

```
->unp profile defaultWLANAccessProfile map service-type spb tag-value 0 isid
1000 bvlan 4000
```

6 Optionally configure the QoS policy list, authentication flag status, or mobile tag status for the “defaultWLANAccessProfile” (changing other parameter values for this profile is not supported). For example:

```
-> unp profile defaultWLANAccessProfile qos-policy-list qlist1
-> unp profile defaultWLANAccessProfile authentication-flag

-> unp profile defaultWLANAccessProfile inactivity-interval 200
ERROR: defaultWLANProfile parameters cannot be modified. Mobile-Tag, Qos-policy-
list, Authentication are only allowed
```

7 Create UNP profiles mapped to SPB service parameters for learning AP client MAC addresses. For example:

```
-> unp profile spb10
-> unp profile spb10 map service-type spb tag-value 10 isid 1010 bvlan 4000

-> unp profile spb20
-> unp profile spb20 map service-type spb tag-value 20 isid 1020 bvlan 4000
```

8 Create UNP classification rules to capture and assign tagged AP client traffic into the UNP service profile configured with a matching VLAN tag value. For example:

```
-> unp classification vlan-tag 10 profile1 spb10
-> unp classification vlan-tag 20 profile1 spb20
```

AP client traffic tagged with SSID VLAN 10 and 20 will be assigned to the “spb10” and “spb20” profiles, respectively. A SAP is then dynamically created based on each profiles service parameters to carry the client traffic through the SPB service domain.

9 Optionally configure the system name, system location, and port alias. The information from one or more of these settings is used to derive the AP Location information that is transmitted by the switch to the connected AP device.

```
-> system name BWIAPS01
-> system location BWI Airport Hotel
-> interfaces port 1/1/15 alias BWI-AP01
```

Verify the OmniSwitch Configuration

1 Use the **show vlan** command to display the VLAN created for AP management and other VLANs created to carry AP client-tagged traffic. Note that “UNP-DYN-VLAN” identifies VLANs dynamically created for AP client traffic. For example:

```
-> show vlan
 vlan    type   admin   oper    ip    mtu          name
------+-------+-------+------+------+------+------------------
1      std       Ena     Ena   Ena    1500    VLAN 1
11     std       Ena     Dis   Dis    1500    VLAN 11-AP Client
200    std       Ena     Dis   Ena    1500    AP Management VLAN
500    unpd      Ena     Dis   Dis    1500    UNP-DYN-VLAN
501    unpd      Ena     Dis   Dis    1500    UNP-DYN-VLAN
```

2 Use the **show vlan members** command to verify the port assignments for the AP management VLAN. For example:

```
-> show vlan 200 members
   port      type        status
----------+-----------+---------------
  1/1/20     qtagged        inactive
  1/1/24     qtagged        inactive
```

3 Use the **show unp profile map** command to verify the AP management VLAN or SPB service is mapped to the “defaultWLANProfile” or “defaultWLANAccessProfile”. For example:

```
-> show unp profile defaultWLANProfile map vlan
Profile Name                      Vlan-Id
--------------------------------+--------
defaultWLANProfile                200

-> show unp profile map service-type spb
Profile                        Tag         Vlan    Mcast   Igmp  Igmp    Mld   Mld
Name                      Isid Value BVlan Xlation  Mode   Snoop Profile Snoop Profile
------------------------+----+-----+-----+-------+-------+-----+-------+-----+-------
spb10                     1010 10    500   Dis     Headend Dis   -       Dis    -
spb20                     1020 20    500   Dis     Headend Dis   -       Dis    -
defaultWLANAccessProfile 1000 0     500   Dis     Headend Dis   -       Dis    -

Total Profile Spb-Map Count: 3
```

4 Use the **show unp profile** command to verify the configurable “defaultWLANProfile” or “defaultWLANAccessProfile” parameter values. For example (the fields highlighted in green are the only configurable parameters for the profile):

```
-> show unp profile defaultWLANProfile
Profile Name: defaultWLANProfile
    Qos Policy      = qlist1,
    Location Policy = -,
    Period Policy   = -,
    CP Profile      = -,
    CP State        = Dis,
    Authen Flag     = Ena,
    Mobile Tag      = Dis,
    SAA Profile     = -,
    Ingress BW      = -,
    Egress BW       = -,
    Ingress Depth   = -,
    Egress Depth    = -,
    Inact Interval  = 10
    Mac-Mobility    = Dis
```

5 Use the **show unp port config** command to make sure Stellar AP devices are connected to UNP bridge or access ports on which the AP mode status is enabled. For example:

```
-> show unp port 1/1/12 config
Port  1/1/12
  Port-Type                      = BRIDGE,
  Redirect Port Bounce           = Disabled,
  802.1x authentication          = Enabled,
  802.1x Pass Alternate Profile  = -,
  802.1x Bypass                  = Disabled,
  802.1x failure-policy          = default,
  Mac-auth allow-eap             = -,
  Mac authentication             = Enabled,
  Mac Pass Alternate Profile     = -,
  Classification                 = Enabled,
  Trust-tag                      = Enabled,
  Default Profile                = -,
  Port Domain Num                = 0,
  AAA Profile                    = -,
  Port Template                  = -,
  Port Control Direction         = Both,
  Egress Flooding                = Not Allowed,
  Admin State                    = Enabled,
  Dynamic Service                = -,
  PVLAN Port Type                = -,
  Force L3-Learning              = Disabled,
  Force L3-Learning Port Bounce  = Disabled,
  AP Mode                        = Enabled,
  802.1x Parameters:
          Tx-Period         = 30,
          Supp-Timeout      = 30,
          Max-req           = 2,

  L2 Profile                     = -

-> show unp port 1/1/15 config
Port  1/4/43
  Port-Type                      = Access,
  802.1x authentication          = Enabled,
  802.1x Pass Alternate Profile  = -,
  802.1x Bypass                  = Disabled,
  802.1x failure-policy          = default,
```

```
  Mac-auth allow-eap              = -,
  Mac authentication              = Enabled,
  Mac Pass Alternate Profile      = -,
  Classification                  = Enabled,
  Trust-tag                       = Disabled,
  Default Profile                 = -,
  Port Domain Num                 = 0,
  AAA Profile                     = -,
  Port Template                   = accessDefaultPortTemplate,
  Admin State                     = Enabled,
  Dynamic Service                 = spb,
  PVLAN Port Type                 = -,
  Force L3-Learning               = Disabled,
  Force L3-Learning Port Bounce   = Enabled,
  AP Mode                         = Enabled,
  802.1x Parameters:
          Tx-Period          = 30,
          Supp-Timeout       = 30,
          Max-req            = 2,

  L2 Profile                      = "ap-SvcUnp"
```

6 Use the **show mvrp configuration** command to verify that MVRP is enabled for the switch. For example:

```
-> show mvrp configuration
MVRP Enabled                     : no,
Maximum VLAN Limit               : 256
```

7 Use the **show system** command and the **show interfaces** command to verify system name, system location, and port alias information. For example:

```
-> show system
System:
  Description:  Alcatel-Lucent Enterprise OS6860E-P48 8.7.1.R01 GA Development,
February 10, 2017.,
  Object ID:    1.3.6.1.4.1.6486.801.1.1.2.1.11.1.8,
  Up Time:      11 days 3 hours 5 minutes and 49 seconds,
  Contact:      Lab Admin,
  Name:         BWIAPS01,
  Location:     BWI Airport hotel,
  Services:     78,
  Date & Time:  THU JAN 22 2020 07:44:07 (UTC)
Flash Space:
    Primary CMM:
      Available (bytes):  1121243136,
      Comments          :  None

-> show interfaces port 1/1/12 alias
 Chas/
 Slot/     Admin     Link        WTR      WTS       Alias
 Port      Status   Status      (sec)    (msec)
--------+----------+---------+----------+----------+-----------------------
 1/1/12    enable     down    0          0          "BWI-AP01"
```

Using L2 GRE Tunneling

Layer 2 Generic Routing Encapsulation (L2 GRE) tunneling is a mechanism that is used to identify and isolate device traffic from the rest of the internal network traffic. This implementation of L2 GRE tunneling is similar to the OmniSwitch implementation of VXLAN as follows:

- L2 GRE tunneling provides a Layer 2 overlay network that is used to tunnel encapsulated traffic over an IP network between two L2 GRE tunnel end points.
- L2 GRE is implemented as a service and can also be associated with a UNP profile.

An L2 GRE tunnel is defined by configuring an L2 GRE end point on a tunnel access switch and an L2 GRE end point on a tunnel aggregation switch.

- Traffic received on the tunnel access switch is classified into a UNP L2 GRE service profile that is mapped to an L2 GRE tunnel service. The profile identifies the device traffic that will be encapsulated with a GRE header and carried over an L2 GRE tunnel to a tunnel aggregation switch.
- When the tunneled traffic reaches the tunnel aggregation switch, the GRE encapsulation is removed and the traffic is then forwarded to a VLAN domain. At this point, the device traffic can gain access to a perimeter network and/or the Internet.

Device traffic received on a UNP port is classified into a UNP profile through any of the Layer 2 or Layer 3 UNP methods for learning and authenticating a user. The final profile obtained from any of these methods must be mapped to an L2 GRE service. The device traffic is then encapsulated and tunneled to the aggregation switch over the L2 GRE tunnel associated with the UNP profile.

This section provides the following information regarding configuring and using the OmniSwitch L2 GRE tunneling mechanism:

- "Configuration Overview and Guidelines" on page 30-113.
- "Quick Steps for Configuring L2 GRE Tunneling" on page 30-119.
- "L2 GRE Tunneling Configuration Example" on page 30-123.

Configuration Overview and Guidelines

The following components comprise the L2 GRE tunneling solution:

- **An L2 GRE tunnel**. A tunnel end point is defined on each edge switch that connects to user devices and on a designated tunnel aggregation switch. These end points define an L2 GRE tunnel through which device traffic is isolated and tunneled over the IP network.
- **Tunnel access switches**. Traffic originates from devices that are connected to a tunnel access switch. The switch then classifies the device traffic into a pre-configured UNP profile. This profile is mapped to an L2 GRE tunnel service. The device traffic is then encapsulated and tunneled through the network to a single tunnel aggregation switch.
- **Tunnel aggregation switch**. All the L2 GRE tunnels originating from all of the tunnel access switches in the network terminate at a tunnel aggregation switch. Device traffic tunneled from the tunnel access switches terminates on a network port for the tunnel aggregation switch. The traffic is then stripped of the GRE encapsulation and passed from the L2 GRE tunnel domain into a VLAN domain.

A switch is operating as an L2 GRE tunnel access switch when a tunnel end point is defined on the switch through the configuration of a UNP service profile. The profile mapping specifies L2 GRE service parameters that are used to dynamically create an L2 GRE service, Service Access Point (SAP), and

Service Distribution Point (SDP) for the tunnel. This is the point at which device traffic will access the L2 GRE tunnel.

A switch is operating as an L2 GRE tunnel aggregation switch when tunnel end points are defined on the switch through the configuration of the necessary L2 GRE service, SAP, and SDP objects. In addition, configuring an external loopback port connection or in-line routing is required on the aggregation switch:

- The external loopback port connection is comprised of an L2 GRE SAP port and a switch port assigned to a VLAN ID. A physical cable connects the two ports together to form the loopback.
- Service-based in-line routing is supported only on the OmniSwitch 9900; an L2 GRE service is bound to an IP interface.
- Front-panel port in-line routing is supported only on the OmniSwitch 6900-V72/C32; a single port is configured with loopback functionality and bound to an L2 GRE SAP and a dedicated VLAN ID.

The L2 GRE tunnel access and tunnel aggregation functionality is supported on the following OmniSwitch platforms:

Switch Platforms	L2GRE Tunnel Access on UNP Bridge Port	L2GRE Tunnel Access on UNP Access Port	L2GRE Tunnel Aggregation
OmniSwitch 6560	Yes	No	No
OmniSwitch 6860	Yes	Yes	Yes
OmniSwitch 6860N	No	Yes	Yes
OmniSwitch 6865	Yes	Yes	Yes
OmniSwitch 6900-T20/T40/X20/X40	No	Yes	Yes
OmniSwitch 6900-X72/Q32	Yes	Yes	Yes
OmniSwitch 6900-V72/C32	No	Yes	Yes
OmniSwitch 6900-X48C6/X48C4E/ T48C6/V48C8/C32E	No	Yes	Yes
OmniSwitch 9900	Yes	Yes	Yes

Notes:

- On an OmniSwitch 6560, a reserved VLAN is required for the L2 GRE tunneling feature.
- Configuring L2 GRE tunneling on an OmniSwitch 6900-T20/T40 or OmniSwitch 6900-X20/X40 is recommended when these models are part of a mixed VC and the L2 GRE SDP is configured on one of the other switches in the VC (for example, on an OmniSwitch 6900-X72/Q32 or OmniSwitch 6900-V72/C32). This ensures that all routing and multicast functionality required to support an L2 GRE tunnel service is limited to a switch that supports SDP configuration.

Additional information and guidelines for using the L2 GRE tunneling mechanism are provided in the following sections:

- “Tunnel Access Switch Configuration Guidelines” on page 30-115.
- “Tunnel Aggregation Switch Configuration Guidelines” on page 30-116.
- “Loopback0 Interface for Tunnel End Points” on page 30-118.

Tunnel Access Switch Configuration Guidelines

Consider the following information and guidelines provided in this section when configuring a switch to operate as an L2 GRE tunnel access switch.

UNP Profile Mapping

- An L2 GRE tunnel end point is defined on an edge switch through the configuration of a UNP profile that is mapped to L2 GRE tunnel service parameter values.
- The L2 GRE service parameter values defined in the profile mapping are used to dynamically create a tunnel service, a SAP, and SDP for an L2 GRE tunnel end point. It is not necessary to manually configure the service, SAP, or SDP to define a tunnel end point on the switch.
- A GRE tunnel Virtual Private Network ID (VPNID), VLAN tag, and a far-end IP address or far-end list name are all required values when configuring the L2 GRE tunnel service mapping for a UNP profile.
- A VPNID serves as a tunnel segment ID and is associated with an L2 GRE service ID.
  - Make sure the VPNID value configured on the tunnel access switch matches the corresponding VPNID value configured on the tunnel aggregation switch.
  - The use of a VPNID is similar to how a VXLAN Network Identifier (VNID) is used to identify a segment of a VXLAN service; the VPNID identifies a segment of an L2 GRE tunnel service and is used in the GRE encapsulation header.
- The far-end IP address should be the IP address of the Loopback0 interface on the L2 GRE tunnel aggregation switch. It is also possible to create an IP address list and specify the name of the list for this parameter value. However, only one IP address is supported at this time.

UNP Port Configuration

Configure the ports on which device traffic will be received as UNP ports. A UNP L2 GRE profile is applied to users learned on UNP bridge and access ports.

- Classifying users learned on UNP access ports into an L2 GRE profile is not supported on the OmniSwitch 6560.
- Classifying users learned on UNP bridge ports into an L2 GRE profile is not supported on the OmniSwitch 6900-T20/T40 or the OmniSwitch 6900-X20/X40.
- Only one UNP L2 GRE profile (one L2 GRE service) can be applied to users learned on UNP bridge ports. The exception to this is on an OmniSwitch 6560 where up to eight UNP L2 GRE profiles (eight L2 GRE services) can be applied to users learned on UNP bridge ports.
- Multiple UNP L2 GRE profiles (multiple L2 GRE services) can be applied to users learned on UNP access ports.
- When users are learned on UNP bridge ports and classified into an L2 GRE profile, the VLAN translation status is not applied. VLAN translation only applies to user traffic learned on UNP access ports.
- The mobile tag functionality is supported on UNP L2 GRE profiles and is applied when users learned on UNP access ports are dynamically assigned to the profile; this functionality is not supported for users learned on UNP bridge ports. Use the **unp profile mobile-tag** command to configure the mobile tag status for the profile.
- There is no VLAN association with the SAP created for the L2 GRE tunnel, so all traffic egressing on the UNP bridge port will be untagged.

Dynamically Created L2 GRE Service Objects

Based on the L2 GRE tunnel service parameters mapped to the UNP profile, an L2 GRE tunnel service, SAP, and SDP are dynamically created when device traffic is classified into the profile. The traffic is then encapsulated and forwarded over the network to the L2 GRE tunnel aggregation switch.

- The dynamic L2 GRE SAP is not based on a VLAN tag; traffic is not mapped to the SAP based on the VLAN tag of the traffic. Instead, the source MAC address of each device assigned to the UNP profile is mapped to the SAP associated with the profile. The SAP serves as a Source Virtual Port (SVP) for all of the MAC addresses assigned to the profile.
- L2 GRE service IDs are dynamically allocated for the UNP profile SAP, similar to how SPB and VXLAN services are dynamically allocated for UNP SPB and VXLAN profile SAPs.
- L2 GRE services do not support multicast modes. By default, all Broadcast, Unknown Unicast, and Multicast (BUM) traffic is replicated by sending a copy to each far-end node over unicast SDPs. This is similar to how the head-end multicast mode works.
- Unicast SDPs to the L2 GRE tunnel aggregation switch are automatically created using the far-end IP address specified in the UNP profile mapping and the reachability of that address in the Layer 3 network. The SDP serves as a Destination Virtual Port (DVP) for all of the MAC addresses assigned to the profile.
- L2 GRE tunneling supports only unicast SDPs; multicast SDPs are not supported. The SDP ID number for unicast SDPs is dynamically allocated.

L2 GRE Reserved VLAN (OmniSwitch 6560)

Configuring a reserved VLAN is required to activate L2 GRE functionality on an OmniSwitch 6560 tunnel access switch. If the reserved VLAN is not created on this switch, then UNP will not learn users in the L2 GRE service domain. A reserved VLAN is not required for other supported OmniSwitch platforms.

Tunnel Aggregation Switch Configuration Guidelines

Consider the following tasks and guidelines provided in this section when configuring a switch to operate as an L2 GRE tunnel aggregation switch.

- Although there can be more than one switch in the network that serves as an L2 GRE tunnel aggregation end point, each edge switch can only connect to one tunnel aggregation switch at a given time. In other words, a tunnel access end point can only be configured to connect to one tunnel aggregation end point.
- There are three methods for defining a tunnel end point on a tunnel aggregation switch: configuring an external loopback port connection, configuring service-based in-line routing (OmniSwitch 9900 only), or configuring front-panel port in-line routing (OmniSwitch 6900-V72/C32 only).
- The L2 GRE service objects, the loopback port connection, and in-line routing are manually configured.

External Loopback Port Connection

When configuring an external loopback port configuration to bridge traffic between the L2 GRE service domain and the VLAN domain, connect one end of a physical loopback cable to a service access port and the other end to a bridge port.

- The access port is assigned to a SAP for the L2 GRE tunnel.
- The bridge port is assigned to a VLAN.

- When GRE tunneled traffic is received on the SAP loopback port, the GRE encapsulation information is removed before the traffic is passed through to the bridge loopback port and forwarded on the VLAN domain.
- When VLAN domain traffic is received on the bridge loopback port, the traffic is passed through to the SAP loopback port, encapsulated, and sent through the GRE tunnel.
- Device traffic enters the L2 GRE tunnel untagged and when the traffic reaches the tunnel aggregation switch, the GRE encapsulation is removed and the traffic is tagged with the VLAN ID of the loopback port to identify the VLAN domain on which the traffic is forwarded.

External Loopback Service Access Port Guidelines

- The loopback service access port and a VLAN tag are used to define the SAP for the L2 GRE tunnel.
- Create an access port Layer 2 profile that will discard all Layer 2 protocol control frames and assign the profile to the loopback service access port.
- Enable VLAN translation on the access port to ensure that egress traffic on the SAP loopback port is tagged. If VLAN translation is not enabled, make sure the VLAN port for the other side of the loopback connection is assigned to the appropriate default VLAN to enable the switching of egress traffic.

External Loopback SAP Guidelines

- The L2 GRE Service Access Point (SAP) is comprised of an L2 GRE service ID associated with the loopback service access port and a VLAN tag encapsulation value. The SAP is used to identify the traffic that will be mapped to the L2 GRE tunnel.
- Specify the VLAN ID of the L2 GRE device traffic VLAN as the encapsulation value for the L2 GRE SAP.
- By default, the trust mode for all SAPs is enabled (802.1p values of the incoming packets are accepted). Do not change this setting for L2 GRE SAPs.

Service-Based In-Line Routing (OmniSwitch 9900)

When configuring service-based in-line routing (OmniSwitch 9900 only) to process and forward traffic in a single pass between the L2 GRE service domain and the VLAN domain, configure an IP interface and bind the interface to an existing L2 GRE service ID.

Device traffic enters the L2 GRE tunnel untagged and when the traffic reaches the tunnel aggregation switch, the GRE encapsulation is removed and the traffic is routed between the L2 GRE service-based IP interface and VLAN-based IP interfaces.

Front-Panel Port In-Line Routing (OmniSwitch 6900-V72/C32)

Configuring in-line routing using a front-panel port (OmniSwitch 6900-V72/C32 only) provides two-pass processing and forwarding of traffic between the L2 GRE service domain and the VLAN domain. This method is similar to the external loopback method except that the loopback function is performed on a single front-panel port (instead of two separate ports) and a physical cable is not required to create the loopback.

The same port is assigned to an IP VLAN and is also configured as a service access port and assigned to an L2 GRE SAP. Traffic is then passed between the L2 GRE service domain and the VLAN domain on the single loopback port.

L2 GRE Service Guidelines

- Make sure the VPNID value configured for an L2 GRE service on the tunnel aggregation switch matches the VPNID value configured for the corresponding L2 GRE service on the tunnel access switch.
- Enable VLAN translation for the L2 GRE service to ensure that egress traffic on the SAP loopback port is tagged; VLAN translation must be enabled at both the access port and service level. If VLAN translation is not enabled, make sure the VLAN port for the other side of the loopback connection is assigned to the appropriate default VLAN to enable the switching of egress traffic.
- The VLAN tag is always removed from ingress packets to ensure that traffic entering the L2 GRE tunnel is untagged.

L2 GRE Tunnel SDP and SDP Binding Guidelines

A Service Distribution Point (SDP) is configured to tunnel L2 GRE traffic to a tunnel access switch. An SDP binding is then configured to bind an L2 GRE service to a unicast SDP.

Only unicast SDPs are supported; this type of SDP is defined by specifying the far-end Loopback0 interface address of the targeted tunnel access switch. This is similar to how SDPs are configured for VXLAN services.

There are two methods available for configuring SDPs and SDP bindings on the tunnel aggregation switch:

- Dynamic configuration achieved through the automatic discovery of remote tunnel end point SDPs for existing L2 GRE services.
  - L2 GRE automatic SDP discovery is enabled by default. The switch discovers the SDPs of remote tunnel end points for a configured L2 GRE service. Once discovered, the switch will dynamically create local SDP and SDP bindings for the remote tunnel end points of the L2 GRE service.
  - The detection of SDPs for remote tunnel end points is based on the traffic received from these end points. If the destination IP address of the remote traffic is the IP address of the local Loopback0 interface and the VPNID of the traffic matches the VPNID of a local L2 GRE service, an SDP and SDP binding is dynamically created for the remote tunnel end point.
- Manual configuration of the necessary SDPs and SDP bindings to the far-end IP address of each remote tunnel access switch (if L2 GRE automatic SDP discovery is disabled).
  - Configure one SDP for each tunnel access switch to which L2 GRE traffic is tunneled. For example, if there are two tunnel access switches, then two SDPs are configured on the tunnel aggregation switch; one SDP for each tunnel access switch.
  - Configure the binding of the L2 GRE service to each SDP to direct one-way GRE encapsulated traffic to the targeted far-end tunnel access switch.

Refer to the configuration steps in the "L2 GRE Tunneling Configuration Example" on page 30-123 for an example of dynamic and manual configuration of SDPs and SDP bindings on an L2 GRE tunnel aggregation switch.

Loopback0 Interface for Tunnel End Points

Configuring a Loopback0 interface IP address is required on each L2 GRE tunnel end point switch.

- The Loopback0 IP address is used as the source IP address for the L2 GRE tunnel.
- On the tunnel access switch, specify the IP address of the Loopback0 interface for the tunnel aggregation switch as the far-end IP address for the UNP profile mapping. The switch will use this IP

address to dynamically create a unicast SDP from the tunnel access switch to the tunnel aggregation switch.

- On the tunnel aggregation switch, specify the IP address of the Loopback0 interface for the tunnel access switch when manually configuring a unicast SDP from the tunnel aggregation switch to the tunnel access switch.

Quick Steps for Configuring L2 GRE Tunneling

This section provides a quick tutorial for configuring the L2 GRE tunneling feature on each participating tunnel access switch and on the tunnel aggregation switch. The configuration steps included in this section are based on the "L2 GRE Tunneling Configuration Example" on page 30-123.

Quick Steps for Configuring an L2 GRE Tunnel Access Switch

The following quick steps are used to configure each switch that will participate in the L2 GRE tunneling configuration as a tunnel access end point switch. An L2 GRE tunnel end point is defined on an access switch by configuring a UNP profile that is mapped to L2 GRE tunnel service parameters.

1 Use the **ip interface** command to configure the Loopback0 interface that will serve as the source IP address for the L2 GRE tunnel.

```
-> ip interface Loopback0 address 10.0.0.1
```

2 Configure other IP interfaces and routes on those interfaces to create a path to the other end of the tunnel (30.0.0.0/24). For example:

```
-> vlan 10
-> vlan 10 members port 1/1/3 tagged

-> ip interface l2-gre-edge-1 address 11.0.0.1 mask 255.255.255.0 vlan 10

-> ip bfd interface "vlan-10"
-> ip bfd interface "vlan-10" admin-state enable
-> ip bfd admin-state enable

-> ip static-route 30.0.0.0/24 gateway 11.0.0.2 bfd-state enable
```

3 Use the **unp profile** command to configure a UNP profile to which device traffic is assigned.

```
-> unp profile Guest
```

4 Use the **unp profile map service-type l2gre** command to map the profile created in Step 2 ("Guest") to L2 GRE service parameters. Specify the Loopback0 interface address of the L2 GRE tunnel aggregation switch as the far-end IP address.

```
-> unp profile Guest map service-type l2gre tag-value 0 vpnid 10 far-end-ip
30.0.0.2
```

5 Use the **unp port-type** command to configure the ports that will connect to user devices as UNP bridge ports.

```
-> unp port 1/1/1-2 port-type bridge
```

Configuring ports as UNP access ports is also supported; UNP bridge ports are used as part of this example configuration.

6 Use the **unp mac-authentication** command to enabled MAC authentication on the UNP bridge ports configured in Step 4.

```
.-> unp port 1/1/1-2 mac-authentication
```

7 Use the **unp default-profile** command to assign the "Guest" profile as the default UNP profile for the ports configured in Step 4. If MAC authentication does not return a profile name, the user device is assigned to the "Guest" profile by default.

```
-> unp port 1/1/1-2 default-profile Guest
```

8 On an OmniSwitch 6560 only, use the **service l2gre reserved-vlan** command to create a reserved VLAN ID for L2 GRE learning.

```
-> service l2gre reserved-vlan 4000
```

Quick Steps for Configuring the L2 GRE Tunnel Aggregation Switch

The following quick steps are used to configure the switch as an L2 GRE tunnel aggregation end point switch. There is only one tunnel aggregation switch, as all L2 GRE tunnels originating on tunnel access switches will terminate to a single tunnel endpoint switch within the network.

1 Use the **ip interface** command to configure the Loopback0 interface that will serve as the source IP address for the L2 GRE tunnel.

```
-> ip interface Loopback0 address 30.0.0.2
```

2 Configure other IP interfaces and routes on those interfaces to create a path to the other end of the tunnel (10.0.0.0/24). For example:

```
-> vlan 30
-> vlan 30 members port 1/1/1 tagged

-> ip interface l2-gre-core-1 address 31.0.0.1 mask 255.255.255.0 vlan 30

-> ip bfd interface "vlan-30"
-> ip bfd interface "vlan-30" admin-state enable
-> ip bfd admin-state enable

-> ip static-route 10.0.0.0/24 gateway 31.0.0.2 bfd-state enable
```

3 Use the **vlan** command to create the VLAN on which device traffic is forwarded to a perimeter network and/or the Internet.

```
-> vlan 50
```

4 Use the **vlan members tagged** command or **vlan members untagged** command to assign a port to the VLAN created in Step 3 (VLAN 50).

```
-> vlan 50 members port 1/1/3 tagged
-> vlan 50 members port 1/1/3 untagged
```

5 Use the **service l2gre** command to create an L2 GRE tunnel service and associate that service with a VPNID, also referred to as an L2 GRE tunnel ID. Make sure the VPNID value specified matches the corresponding VPNID value that was configured on the tunnel access switch.

```
-> service 100 l2gre vpnid 10 vlan-xlation enable remove-ingress-tag enable
admin-state enable description "Guest Service"
```

6 The following steps are used to define an external loopback port configuration (see Step 7 and Step 8 to define an in-line routing configuration).

a. Use the **service l2profile** command to configure a Layer 2 profile to drop all L2 protocol control frames. This profile is assigned to the access port that will serve as the SAP loopback port.

```
-> service l2profile Guest-l2profile stp drop 802.1x drop 802.1ab drop 802.3ad
drop gvrp drop mvrp drop amap drop
```

b. Use the **service access** command to configure a port as a service access port and assign a Layer 2 profile to the same port. Specify the port that will serve as the SAP loopback port and the Layer 2 profile created in the previous step ("Guest-l2profile").

```
-> service access port 1/1/2 l2profile Guest-l2profile vlan-xlation enable
description "Guest Loopback Port"
```

c. Use the **service sap** command to create a Service Access Point (SAP) by associating the L2 GRE tunnel service created in Step 5 (service 100) with the access port (1/1/2) and the VLAN ID created in Step 3 (VLAN 50).

```
-> service 100 sap port 1/1/2:50 trusted admin-state enable description "Guest
SAP VLAN 50"
```

d. Connect one end of the loopback cable to the port that was assigned to VLAN 50 in Step 4 (bridge port 1/1/3) and the other end of the cable to the SAP port (access port 1/1/2).

7 The following steps are used to define a service-based in-line routing configuration (OmniSwitch 9900 only).

a. Use the **ip interface** command to define an IP address for the VLAN created in Step 3 (VLAN 50).

```
-> ip interface l2gre-vlan50 address 41.0.0.1 vlan 50
```

b. Use the **ip interface** command to define an IP address for the L2 GRE service created in Step 5 (service 100).

```
-> ip interface l2gre-service100 address 51.0.0.1 service 100
```

Binding an IP interface to an L2 GRE service is done instead of configuring a VLAN-to-SAP external loopback port connection. L2 GRE tunnel traffic is routed between the service-based IP interface and the VLAN-based IP interfaces.

8 The following steps are used to define a front-panel port in-line routing configuration (OmniSwitch 6900-V72/C32 only).

a. Use the **interfaces** command with the **loopback** option to configure a port to operate in the loopback mode. For example:

```
-> interfaces port 1/1/2 loopback
```

a. Use the **service l2profile** command to configure a Layer 2 profile to drop all L2 protocol control frames. This profile is assigned to the access port that will serve as the SAP loopback port.

```
-> service l2profile Guest-l2profile stp drop 802.1x drop 802.1ab drop 802.3ad
drop gvrp drop mvrp drop amap drop
```

b. Use the **service access** command to configure loopback port 1/1/2 as a service access port with VLAN translation enabled and assign the Layer 2 profile ("Guest-l2profile") to that port. Loopback port 1/1/2 will serve as the SAP loopback port.

```
-> service access port 1/1/2 l2profile Guest-l2profile vlan-xlation enable
description "Guest Loopback Port"
```

c. Use the **service sap** command to create a Service Access Point (SAP) by associating the L2 GRE tunnel service created in Step 5 (service 100) with loopback port 1/1/2. Specify the dedicated VLAN to which port 1/1/2 will be assigned (see Step 8d) as the SAP encapsulation value.

```
-> service 100 sap port 1/1/2:400 trusted admin-state enable description "Guest
SAP VLAN 400"
```

d. Use the **ip interface rtr-port** command with the **tagged vlan** option to assign loopback port 1/1/2 to an IP interface. This command also creates the specified VLAN ID and tags the loopback port with that VLAN. For example:

```
-> ip interface "L2GRE Loopback" address 100.1.1.1/8 rtr-port port 1/1/2 tagged
vlan 400
```

Configuring the loopback function on port 1/1/2 and assigning that port to both an L2 GRE SAP and a dedicated IP interface VLAN (VLAN 400), provides two-pass in-line routing. L2 GRE traffic is passed between the service domain and VLAN domain on the same port.

9 Configure a Service Distribution Point (SDP) and SDP binding from the L2 GRE tunnel aggregation switch to each L2 GRE tunnel access switch. There are two methods available for doing this: dynamic configuration through automatic SDP discovery of remote tunnel SDPs for existing L2 GRE services or manual configuration (if automatic discovery is disabled).

a. The dynamic configuration method is enabled by default and automatically creates a unicast SDP and SDP binding to each remote tunnel end point that is discovered by the tunnel aggregation switch. If this method is disabled, use the **service l2gre auto-discover** command to enable automatic discovery.

```
-> service l2gre auto-discover admin-state enable
```

b. If dynamic configuration is disabled, manually configure an SDP to the far-end IP address (Loopback0 address) of each tunnel access switch using the **service sdp l2gre** command and use the **service bind-sdp** command to bind each SDP to an L2 GRE service.

```
-> service sdp 20 l2gre far-end 10.0.0.1 admin-state enable description "Guest
SDP 10.0.0.1"
-> service sdp 21 l2gre far-end 20.0.0.1 admin-state enable description "Guest
SDP 20.0.0.1"
-> service 100 bind-sdp 20 21
```

L2 GRE Tunneling Configuration Example

The use case described in this section provides an example of using L2 GRE tunneling to identify and isolate guest traffic coming into the network. Device traffic is identified as guest traffic through UNP classification on the tunnel access switch. The guest traffic is encapsulated with a GRE header and tunneled through the network to a tunnel aggregation switch. When the guest traffic reaches the tunnel aggregation switch, the GRE encapsulation is removed and the traffic is then forwarded through the "Guest" VLAN to the Internet. Access to all the network resources, such as DHCP and DNS, is also provided through the L2 GRE tunnel for the guest traffic.

The L2 GRE tunnel is defined through the configuration of tunnel end points on the tunnel access switch and on the tunnel aggregation switch. There are three options for defining a tunnel end point on a tunnel aggregation switch: an external loopback configuration, a service-based in-line routing configuration, or a front-panel port in-line routing configuration. This use case example includes an external loopback configuration, as shown in the following diagram:

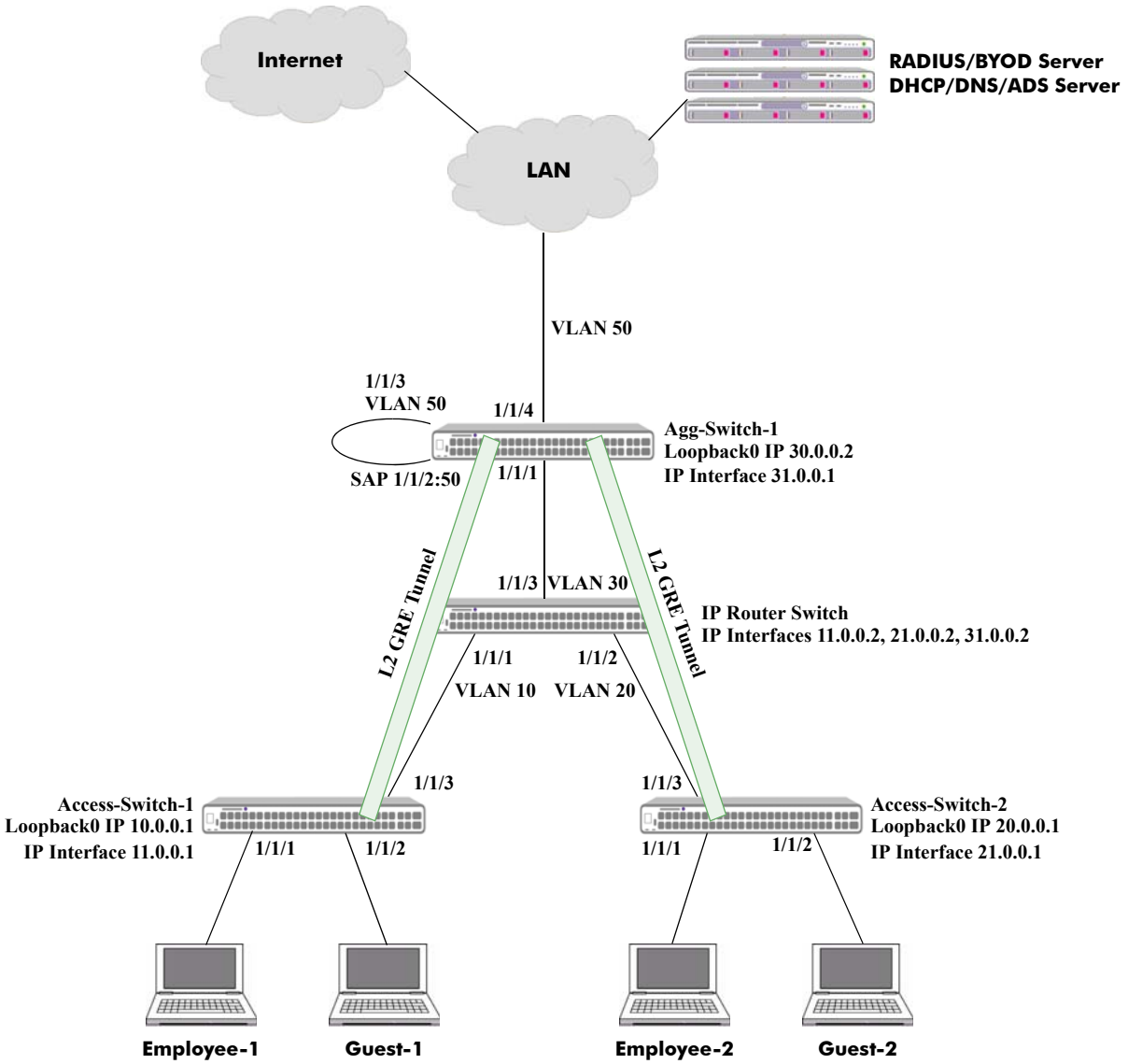


Figure 30-9 : OmniSwitch L2 GRE Tunneling Configuration Example

In this configuration example:

- A UNP Guest profile is configured on Access-Switch-1 and Access-Switch-2. The Guest profile is mapped to an L2 GRE tunnel service.
- Traffic from the Guest-1 and Guest-2 devices is classified into the UNP Guest profile, encapsulated, and then tunneled through the L2 GRE tunnel to Agg-Switch-1.
- When the encapsulated guest traffic reaches Agg-Switch-1, the GRE encapsulation information is removed and the traffic is passed through the SAP loopback port to the VLAN loopback port.
- The VLAN loopback port is tagged with VLAN 50, where the guest traffic is then granted access to perimeter network resources and the Internet.
- Return traffic destined for Guest-1 and Guest-2 is forwarded on VLAN 50 and then passed through the VLAN loopback port to the SAP loopback port.
- The SAP loopback port is mapped to an L2 GRE tunnel service, where the guest traffic is encapsulated and then sent back through the tunnel to the appropriate tunnel access switch.
- When the encapsulated guest traffic reaches the intended tunnel access switch, the GRE encapsulation information is removed and the traffic is forwarded to the guest device.
- A UNP Employee profile is configured on Access-Switch-1 and Access-Switch-2. The Employee profile is mapped to a VLAN.
- Traffic from the Employee-1 and Employee-2 devices is classified into the UNP Employee profile and forwarded on the mapped VLAN through the network.

The following command examples further illustrate how the Guest Tunneling functionality is configured on each L2 GRE tunnel access switch and on the L2 GRE tunnel aggregation switch.

Access-Switch-1:

```
-> ip interface “Loopback0” 10.0.0.1

-> vlan 10
-> vlan 10 members port 1/1/3 tagged

-> ip interface l2-gre-edge-1 address 11.0.0.1 mask 255.255.255.0 vlan 10

-> ip bfd interface "vlan-10"
-> ip bfd interface "vlan-10" admin-state enable
-> ip bfd admin-state enable

-> ip static-route 30.0.0.0/24 gateway 11.0.0.2 bfd-state enable

-> unp profile Guest
-> unp profile Guest map service-type l2gre tag-value 0 vpnid 10 far-end-ip
30.0.0.2
-> unp port 1/1/2 port-type bridge
-> unp port 1/1/2 mac-authentication
-> unp port 1/1/2 default-profile Guest

-> vlan 40
-> unp profile Employee
-> unp profile Employee map vlan 40
-> unp port 1/1/1 port-type bridge
-> unp port 1/1/1 mac-authentication
-> unp port 1/1/1 default-profile Employee
```

Access-Switch-2:

```
-> ip interface "Loopback0" 20.0.0.1

-> vlan 20
-> vlan 20 members port 1/1/3 tagged

-> ip interface l2-gre-edge-2 address 21.0.0.1 mask 255.255.255.0 vlan 20

-> ip bfd interface "vlan-20"
-> ip bfd interface "vlan-20" admin-state enable
-> ip bfd admin-state enable

-> ip static-route 30.0.0.0/24 gateway 21.0.0.2 bfd-state enable

-> unp profile Guest
-> unp profile Guest map service-type l2gre tag-value 0 vpnid 10 far-end-ip
30.0.0.2
-> unp port 1/1/2 port-type bridge
-> unp port 1/1/2 mac-authentication
-> unp port 1/1/2 default-profile Guest

-> vlan 40
-> unp profile Employee
-> unp profile Employee map vlan 40
-> unp port 1/1/1 port-type bridge
-> unp port 1/1/1 mac-authentication
-> unp port 1/1/1 default-profile Employee
```

Agg-Switch-1 (with manual configuration of SDP and SDP bindings):

```
-> ip interface "Loopback0" 30.0.0.2

-> vlan 30
-> vlan 30 members port 1/1/1 tagged

-> ip interface l2-gre-core-1 address 31.0.0.1 mask 255.255.255.0 vlan 30

-> ip bfd interface "vlan-30"
-> ip bfd interface "vlan-30" admin-state enable
-> ip bfd admin-state enable

-> ip static-route 10.0.0.0/24 gateway 31.0.0.2 bfd-state enable
-> ip static-route 20.0.0.0/24 gateway 21.0.0.2 bfd-state enable

-> vlan 50
-> vlan 50 members port 1/1/3 tagged
-> vlan 50 members port 1/1/4 untagged

-> service l2profile Guest-l2profile stp drop 802.1x drop 802.1ab drop 802.3ad drop
gvrp drop mvrp drop amap drop
-> service access port 1/1/2 l2profile Guest-l2profile vlan-xlation enable
description "Guest Loopback Port"

-> service 100 l2gre vpnid 10 vlan-xlation enable remove-ingress-tag enable admin-
state enable description "Guest Service"
-> service 100 sap port 1/1/2:50 trusted admin-state enable description "Guest SAP
VLAN 50"
```

```
-> service sdp 20 l2gre far-end 10.0.0.1 admin-state enable description "Guest SDP
10.0.0.1"
-> service sdp 21 l2gre far-end 20.0.0.1 admin-state enable description "Guest SDP
20.0.0.1"

-> service 100 bind-sdp 20 21
```

Agg-Switch-1 (with automatic discovery of remote tunnel SDPs):

```
-> ip interface "Loopback0" 30.0.0.2

-> vlan 30
-> vlan 30 members port 1/1/1 tagged

-> ip interface l2-gre-core-1 address 31.0.0.1 mask 255.255.255.0 vlan 30

-> ip bfd interface "vlan-30"
-> ip bfd interface "vlan-30" admin-state enable
-> ip bfd admin-state enable

-> ip static-route 10.0.0.0/24 gateway 31.0.0.2 bfd-state enable
-> ip static-route 20.0.0.0/24 gateway 21.0.0.2 bfd-state enable

-> vlan 50
-> vlan 50 members port 1/1/3 tagged
-> vlan 50 members port 1/1/4 untagged

-> service l2profile Guest-l2profile stp drop 802.1x drop 802.1ab drop 802.3ad drop
gvrp drop mvrp drop amap drop
-> service access port 1/1/2 l2profile Guest-l2profile vlan-xlation enable
description "Guest Loopback Port"

-> service 100 l2gre vpnid 10 vlan-xlation enable remove-ingress-tag enable admin-
state enable description "Guest Service"
-> service 100 sap port 1/1/2:50 trusted admin-state enable description "Guest SAP
VLAN 50"
```

If automatic discovery is enabled (the default) for this example, remote tunnel SDPs are discovered and local SDP and SDP bindings are dynamically configured as follows:

- Agg-Switch-1 receives encapsulated L2 GRE traffic from Access-Switch-1 and Access-Switch-2 with a destination IP set to 30.0.0.2 and a VPNID of 10.
- Agg-Switch-1 identifies the tunneled traffic as coming from a remote L2 GRE tunnel end point and dynamically creates an SDP to 10.0.0.1 for Access-Switch-1 and an SDP to 20.0.0.1 for Access-Switch-2.
- The dynamically created SDPs are then bound to L2 GRE service 100, which is assigned to VPNID 10.

IP Router Switch:

```
-> vlan 10
-> vlan 10 members port 1/1/1 tagged

-> vlan 20
-> vlan 20 members port 1/1/2 tagged

-> vlan 30
-> vlan 30 members port 1/1/3 tagged
```

```
-> ip interface to-l2-gre-edge-1 address 11.0.0.2 mask 255.255.255.0 vlan 10
-> ip interface to-l2-gre-edge-2 address 21.0.0.2 mask 255.255.255.0 vlan 20
-> ip interface to-l2-gre-core address 31.0.0.2 mask 255.255.255.0 vlan 30

-> ip bfd interface "vlan-10"
-> ip bfd interface "vlan-10" admin-state enable
-> ip bfd interface "vlan-20"
-> ip bfd interface "vlan-20" admin-state enable
-> ip bfd interface "vlan-30"
-> ip bfd interface "vlan-30" admin-state enable
-> ip bfd admin-state enable

-> ip static-route 10.0.0.0/24 gateway 11.0.0.1 bfd-state enable
-> ip static-route 20.0.0.0/24 gateway 21.0.0.1 bfd-state enable
-> ip static-route 30.0.0.0/24 gateway 31.0.0.1 bfd-state enable
```

Using Quarantine Manager and Remediation

A client MAC address is determined to be in a quarantined state when one of the following occurs:

- The OmniVista Quarantine Manager (OVQM) application receives a TRAP indicating that the MAC address has to be quarantined. The TRAP may come from a network anomaly detection application or from an IDS running in the same subnet as the client.
- A list containing the quarantined MAC address is manually configured on OVQM.
- A list containing the quarantined MAC address is manually configured on every switch in the network.

After the list of quarantined MAC addresses is known, OVQM can add these addresses to the Quarantine MAC group and push the configuration to the switches in a logical group or to all switches.

The Access Guardian Quarantine Manager and Remediation (QMR) feature moves the users associated with the quarantined MAC addresses to a QMR restricted role. A built-in policy list is associated with the QMR role that restricts quarantined users to communicating with a designated remediation server until their quarantined status is corrected.

QMR works on UNP (bridge and access) ports and on non-UNP ports. Layer 2 source learning receives the quarantined MAC addresses from QMR and changes the MAC address status to “quarantined”. The **show mac-learning** and **show unp user** commands display the appropriate status for quarantined MAC addresses.

The following QMR components are configured through QoS and Access Guardian CLI commands:

- **Quarantined MAC address group**. The Access Guardian configures the name of the Quarantine MAC group on the OmniSwitch. This MAC address group contains the MAC addresses of users that are quarantined and are candidates for remediation.

  The default name of the MAC group is "Quarantined”, but the name can be changed using the **qos quarantine mac-group** command. For example:

  ```
  -> qos quarantine mac-group badMacs
  ```

- **Remediation server URL**. The Access Guardian **qmr quarantine path** command is used to specify a URL to which users are redirected for remediation. For example:

  ```
  -> qmr quarantine path www.qmr.ale.com
  ```

  Redirecting quarantined users learned on UNP access ports for remediation is not supported; users learned on UNP bridge ports can be redirected for remediation.

- **Remediation server and exception subnets**. When a client is quarantined, all the traffic from the client is blocked by default. However, the administrator can configure access to some exception subnets to which the quarantined client can be redirected, such as the IP address of a remediation server to obtain updates and correct its quarantined state.

  The **qmr quarantine allowed-name** command is used to designate IP addresses that a quarantined client can access. For example:

  ```
  -> qmr quarantine allowed-name it-helpdesk 10.1.1.0 ip-mask 255.255.255.0
  ```

  Make sure the IP address for the remediation server is included in the allowed list of subnets. Specifying a maximum of two exception subnets is allowed.

- **Quarantined Page**. When a client is quarantined and a remediation server URL is not configured, QMR can send a Quarantine Page to notify the client of its quarantined state. To enable or disable the

sending of a Quarantine Page, use the **qmr quarantine page** command. For example:

```
-> qmr quarantine page enable
```

- **QMR custom proxy port**. This specifies the HTTP proxy port number to which quarantined client traffic is redirected for remediation. The default HTTP port used is TCP 80 and TCP 8080.

  The **qmr quarantine custom-proxy-port** command is used to configure a different proxy port number to use. For example:

```
-> qmr quarantine custom-proxy 8888
```

Use the **show qmr** command to verify the QMR configuration on the switch and use the **show quarantine mac group** command to display a list of the quarantined MAC addresses known to the switch.

For an example of configuring a custom QMR role (policy list) to apply to quarantined users, see the "Application Example 6: Restricted Role (Policy List) Assignment" on page 30-141.

Access Guardian Application Examples

This section provides some typical application examples in which Access Guardian is used to implement network access control in a sample network configuration. The following diagram depicts an Access Guardian network implementation that applies to all of the application examples in this section.

For application examples of the OmniSwitch Bring Your Own Devices (BYOD) solution provided through Access Guardian with the Unified Policy Access Manager (UPAM) or the ClearPass Policy Manager (CPPM), see the "BYOD Application Examples" on page 30-180.

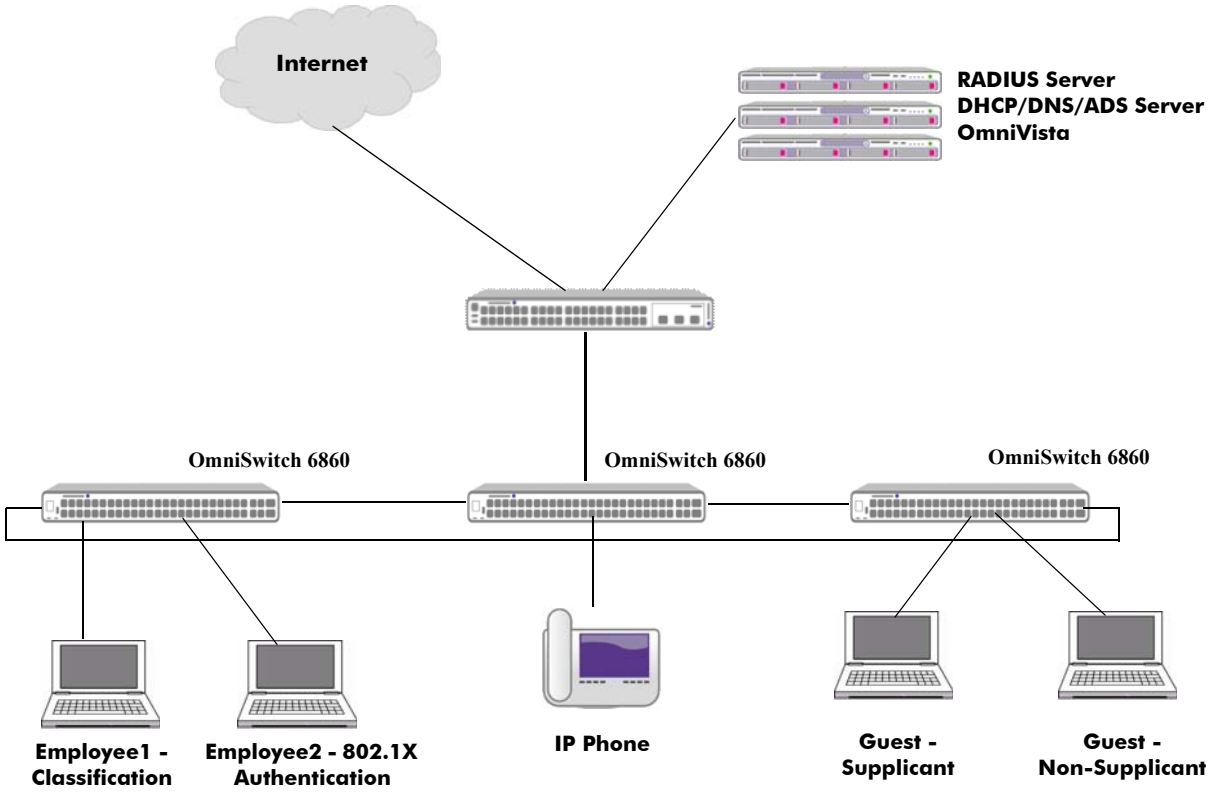


Figure 30-10 : Access Guardian Network Configuration Example

The following application examples are provided in this section:

- "Application Example 1: Classification (Port Mobility)" on page 30-131
- "Application Example 2: 802.1X Authentication" on page 30-132
- "Application Example 3: Internal Captive Portal Authentication" on page 30-134
- "Application Example 4: Supplicant/Non-supplicant with Captive Portal Authentication" on page 30-136
- "Application Example 5: IP Phone (LLDP Network Policy TLV/Mobile Tag)" on page 30-139
- "Application Example 6: Restricted Role (Policy List) Assignment" on page 30-141

Application Example 1: Classification (Port Mobility)

In this configuration example, network access control for Employee1 is provided through the Access Guardian classification mechanism; no authentication is necessary. Classification is a function of the UNP feature and is enabled or disabled on UNP ports. Once enabled, the port and devices connected to the port are eligible for dynamic assignment to a VLAN that is associated with a UNP profile.

To determine which UNP profile a device is assigned to, the administrator configures UNP classification rules and assigns those rules to the appropriate profile. When traffic received on a UNP port that has classification enabled matches the criteria of a specific classification rule, the user device is moved into the profile associated with the rule and assigned to the VLAN associated with the profile.

This application example uses a MAC address range classification rule to dynamically assign Employee1 into VLAN 20. The following steps provide a brief tutorial for how to configure this example:

1 Create the required VLANs.

```
-> vlan 10 admin-state disable name vlan10-block
-> vlan 20 admin-state enable name vlan20-corporate
```

2 Create the required UNP profile and map the profile to VLAN 20.

```
-> unp profile corporate
-> unp profile corporate map vlan 20
```

3 Create another UNP profile that will serve as a default profile; map the profile to VLAN 10.

```
-> unp profile def_unp
-> unp profile def_unp map vlan 10
```

4 Create a MAC range classification rule and associate the rule to the “corporate” UNP profile.

```
-> unp classification-rule rule1 mac-address-range 08:00:27:00:98:0A
08:00:27:00:98:FF profile1 corporate
```

5 Enable UNP on ports that will connect to user devices.

```
-> unp port 1/1/1 port-type bridge
```

6 Set the default UNP profile on the port.

```
-> unp port 1/1/1 default-profile def_unp
```

7 Enable classification on the UNP port.

```
-> unp port 1/1/1 classification
```

How it Works

In this example, traffic received on the UNP port triggers the following classification process:

- Device traffic is examined and matched against all UNP classification rules.
- If the MAC address of a user device is within the range of MAC addresses specified in the MAC address range rule, the user is classified into the “corporate” profile and assigned to VLAN 20.
- If the MAC address of a user is not within the MAC address range and does not match any other UNP classification rules on the switch, then the user is classified into the “def_unp” profile and assigned to VLAN 10.

- The MAC addresses are learned in the assigned VLANs and the device port is now an untagged member of the assigned VLANs.

UNP Port Template Example

In Application Example 1 (Classification), individual CLI commands are used in Steps 6 and 7 to configure UNP port parameters. However, it is possible to create a UNP port template that defines port configuration parameters and assigns these parameters to a template name.

1 Create a port template.

```
-> unp port-template classify-template
```

2 Configure the template to enable classification on the associated UNP port.

```
-> unp port-template classify-template classification
```

3 Configure the template to assign a default UNP profile to the UNP port.

```
-> unp port-template classify-template default-profile def_unp
```

4 Assign the template to the UNP port to apply the template configuration.

```
-> unp port 1/1/1 port-template classify-template
```

Using a port template to configure UNP ports helps to simplify and expedite the configuration process. Templates allow the administrator to easily replicate a specific configuration across multiple UNP ports.

Application Example 2: 802.1X Authentication

In this example, network access control for Employee2 is provided through the Access Guardian 802.1X authentication mechanism. Authentication is a function of the UNP feature and is enabled or disabled on UNP ports. There are two types of authentication supported at the port (Layer 2) level: 802.1X and MAC authentication.

This application example demonstrates the 802.1X authentication capability for a supplicant (802.1X) device. The following steps provide a brief tutorial for how to configure this example:

1 Configure a server as a RADIUS server on the switch.

```
-> aaa radius-server alu-authserver host 10.242.254.101 hash-key secret
retransmit 3 timeout 2 auth-port 1812 acct-port 1813
```

2 Configure the switch to use “alu-authserver” (identified in Step 1) for 802.1X device authentication.

```
-> aaa device-authentication 802.1x alu-authserver
```

3 Configure the switch to use “alu-authserver” for RADIUS server accounting.

```
-> aaa accounting 802.1x alu-authserver
```

4 Create the required VLANs.

```
-> vlan 10 admin-state disable name vlan10-block
-> vlan 20 admin-state enable name vlan20-corporate
```

5 Create the required UNP profile and map the profile to VLAN 20.

```
-> unp profile corporate
-> unp profile corporate map vlan 20
```

6 Create another UNP profile that will serve as a default profile; map the profile to VLAN 10.

```
-> unp profile def_unp
-> unp profile def_unp map vlan 10
```

7 Create an edge template to apply UNP port configuration parameters.

```
-> unp port-template 802.1X-template
```

8 Configure the template to enable 802.1X authentication and define an alternate UNP profile to use if the RADIUS server does not return a UNP profile name upon successful authentication.

```
-> unp port-template 802.1x-template 802.1x-authentication
-> unp port-template 802.1x-template 802.1x-authentication pass-alternate
corporate
```

9 Assign the port template to a UNP port.

```
-> unp port 2/1/1 port-template 802.1x-template
```

How it Works

In this example, traffic received on the UNP port will trigger the following device authentication process on the switch:

- Supplicant device traffic will trigger 802.1 x authentications first.
- If 802.1X authentication passes, the client is classified into to the “corporate” UNP profile and assigned to VLAN 20 or classified into the UNP profile returned from RADIUS server.
- If 802.1X authentication fails and classification is not enabled and a default profile is not assigned, the MAC address of the user device is filtered (blocked).
- In this example, MAC authentication and classification are not enabled on the UNP port, so neither MAC authentication or classification will be triggered for a non-supplicant device. However, a default UNP profile is configured for the port, so a non-supplicant device will get classified into that profile.

AAA Profile Example

In Application Example 2 (802.1X Authentication), individual CLI commands are used in Steps 1–3 to configure AAA parameters. However, it is possible to create an AAA profile that defines the AAA server configuration parameters and assigns these parameters to a profile name. The profile is then assigned to a UNP port or a UNP port template.

1 Configure the AAA profile name.

```
-> aaa profile ag-aaa-profile
```

2 Configure the profile to specify the “alu-authserver” for 802.1X device authentication.

```
-> aaa profile ag-aaa-profile device-authentication 802.1x “alu-authserver”
```

3 Configure the profile to specify the “alu-authserver” for RADIUS server accounting.

```
-> aaa profile ag-aaa-profile accounting 802.1x alu-authserver
```

4 Assign the AAA profile to a UNP port or to a UNP port template.

```
-> unp port 2/1/1 aaa-profile ag-aaa-profile
-> unp port-template 802.1x-template aaa-profile ag-aaa-profile
```

Application Example 3: Internal Captive Portal Authentication

In this example, network access control is provided for different types of users through Access Guardian internal Captive Portal authentication. For example, university students, teachers, and visitors authenticating through Captive Portal to receive different QoS policy lists based on the their role in the network.

Internal Captive Portal authentication is initiated only through a UNP profile. As a result, the user must initially be classified into a profile through Layer 2 authentication (802.1X or MAC), rule classification, or assigned to a default UNP profile.

The UNP profile assigned must have Captive Portal authentication enabled. The Captive Portal authentication process is used to assign a network access role (QoS policy list) to the user. Different policy lists may be assigned to different users.

This application example demonstrates the internal Captive Portal authentication capability to dynamically assign a network access role for a user device. The following steps provide a brief tutorial for how to configure this example.

Network Configuration for Captive Portal Support

1 Configure the network DHCP server to give out the IP addresses in the subnet of the VLAN associated with the UNP profile that will be used for Captive Portal authentication.

2 Configure the DNS with a DNS entry to map the Captive Portal name to the Captive Portal IP address that is configured on the switches in the network.

OmniSwitch Configuration for Captive Portal Support

1 Configure a RADIUS server.

```
-> aaa radius-server alu-authserver host 10.242.254.101 hash-key secret
retransmit 3 timeout 2 auth-port 1812 acct-port 1813
```

2 Create an AAA profile to pre-define and apply a specific AAA configuration for this example.

```
-> aaa profile ag-aaa-profile
-> aaa profile ag-aaa-profile device-authentication captive-portal alu-
authserver
-> aaa profile ag-aaa-profile accounting captive-portal alu-authserver
```

3 Create the required VLANs.

```
-> vlan 10 admin-state disable name vlan-block
-> vlan 30 admin-state enable name vlan-guest
```

4 Create the QoS policy list to apply to the user upon successful Captive Portal authentication.

```
-> policy condition cp-default-C1 source ip Any destination ip Any
-> policy action cp-default-A1
-> policy rule cp-default-R1 condition cp-default-C1 action cp-default-A1
-> policy list cp-default-list type unp
-> policy list cp-default-list rules cp-default-R1
-> qos apply
```

5 Create a UNP profile named “guest”.

```
-> unp profile guest
```

6 Map the UNP profile to an appropriate VLAN.

```
-> unp profile guest map vlan 30
```

7 Create another UNP profile to use as a default profile for a UNP port.

```
-> unp profile def_unp
```

8 Map the UNP profile to VLAN 10.

```
-> unp profile def_unp map vlan 10
```

9 Create a UNP port template to pre-define and apply configuration parameters to the UNP port.

```
-> unp port-template cp-only-template
```

10 Set the default profile parameter for the port template to “guest”.

```
-> unp port-template cp-only-template default-profile guest
```

Setting the default UNP profile to “guest” will move the clients into VLAN 30 first. Then a policy list received through Captive Portal authentication is applied to the user (overriding the policy list, if any, that was previous applied through the “guest” profile).

11 Assign the UNP port template to a UNP port.

```
-> unp port 1/1/2 port-template cp-only-template
```

12 Create a Captive Portal profile to pre-define and apply Captive Portal configuration parameters.

```
-> captive-portal-profile cp-profile
-> captive-portal-profile cp-profile aaa-profile ag-aaa-profile
```

13 Set the Captive Portal profile parameters for the authentication pass policy list and the success URL. The Captive Portal IP address is set to 10.123.0.1 by default.

```
-> captive-portal-profile cp-profile authentication-pass policy-list cp-default
-list
-> captive-portal-profile cp-profile success-redirect-url http://test-cp.com/
success.html
```

The authentication pass parameter specifies the policy list to apply to the device if Captive Portal authentication is successful but the RADIUS server does not return a policy list. The success URL specifies where to redirect the user device after a successful Captive Portal authentication attempt.

14 Enable Captive Portal for the UNP profile and assign the Captive Portal profile to the same UNP profile.

```
-> unp profile guest captive-portal-authentication
-> unp profile guest captive-portal-profile cp-profile
```

How it Works

In this example, traffic arriving on the UNP port triggers the following process on the switch:

- Authentication and classification are disabled on the UNP port, so the client is assigned to the default UNP profile and associated VLAN.
- Because the default UNP profile (associated with the port template assigned to the UNP port) is enabled for Captive Portal authentication, the Captive Portal authentication process is triggered.
- The client is placed into a built-in Captive Portal pre-login role which does the following:
  - Allows the client network access only for DHCP, DNS, ARP, and ICMP traffic.
  - Traps and redirects client HTTP/HTTPS traffic to the internal Captive Portal Web server on the switch. The Captive Portal server name and IP address was resolved by the client through DNS.
  - Client is presented with an internal Captive Portal login page.
  - Client enters user credentials which are then authenticated through the RADIUS server designated for Captive Portal authentication.
- Successful Captive Portal authentication results in the assignment of a policy list that was returned from the RADIUS server or specified through the Captive Portal authentication pass configuration.
- The client remains in the “guest” UNP profile assigned to VLAN 30 and is presented with a Captive Portal login status page.
- If Captive Portal authentication fails, the client remains in the built-in Captive Portal pre-login role.

Application Example 4: Supplicant/Non-supplicant with Captive Portal Authentication

In this example, network access control is provided for corporate devices and guest devices trying to access the network on the same port. The scenarios covered in this example are as follows:

- Corporate supplicant device.
  - Passes 802.1X authentication.
  - Is assigned a UNP corporate profile with an associated VLAN.
- Corporate user with non-supplicant, non-corporate device.
  - Does not trigger 802.1X authentication.
  - Fails MAC authentication.
  - When MAC authentication fails and classification is not enabled, a default UNP profile associated with the UNP port will be assigned. Captive Portal authentication is enabled for the default profile.
  - The Captive Portal authentication pass condition may apply a new access policy list or the access policy list associated with the default profile is applied.

- Guest supplicant device.
  - Fails 802.1X authentication.
  - If an 802.1X failure policy is not set and classification is not enabled, a default UNP profile associated with the UNP port will be assigned. Captive Portal authentication is enabled for the default profile.
  - The Captive Portal authentication pass condition may apply a new access policy list or the access policy list associated with the default profile is applied.
- Guest non-supplicant device.
  - Fails 802.1X authentication.
  - MAC authentication is not automatically triggered, unless explicitly enabled on the UNP port.
  - If MAC authentication fails and classification is not enabled, a default UNP profile associated with the UNP port will be assigned. Captive Portal authentication is enabled for the default profile.
  - The Captive Portal authentication pass condition may apply a new access policy list or the access policy list associated with the default profile is applied.

The following steps provide a brief tutorial for how to configure this application example:

1 Configure a RADIUS server.

```
-> aaa radius-server alu-authserver host 10.242.254.101 hash-key secret
retransmit 3 timeout 2 auth-port 1812 acct-port 1813
```

2 Create an AAA profile to pre-define and apply a specific AAA configuration for this example.

```
-> aaa profile ag-aaa-profile device-authentication 802.1x alu-authserver
-> aaa profile ag-aaa-profile accounting 802.1x alu-authserver
-> aaa profile ag-aaa-profile device-authentication mac alu-authserver
-> aaa profile ag-aaa-profile accounting mac alu-authserver
-> aaa profile ag-aaa-profile device-authentication captive-portal alu-
authserver
-> aaa profile ag-aaa-profile accounting captive-portal alu-authserver
```

3 Create the required VLANs.

```
-> vlan 10 admin-state disable name vlan-block
-> vlan 20 admin-state enable name vlan-corporate
-> vlan 30 admin-state enable name vlan-guest
```

4 Create the required UNP profiles.

```
-> unp profile corporate
-> unp profile guest
```

5 Map the UNP profiles to the appropriate VLANs.

```
-> unp profile corporate map vlan 20
-> unp profile guest map vlan 30
```

6 Create a default UNP profile.

```
-> unp profile def_unp
```

7 Map the default UNP profile to VLAN 10.

```
-> unp profile def_unp map vlan 10
```

8 Create a port template to pre-define and apply configuration parameters to the UNP port.

```
-> unp port-template auth-template
```

9 Set the default UNP profile parameter for the port template to “guest”.

```
-> unp port-template auth-template default-profile guest
```

10 Set the MAC and 802.1X authentication parameters to “enable” for the port template. Can also define a pass alternate UNP profile for the template in case the RADIUS server does not return a UNP profile name when 802.1X or MAC authentication passes.

```
-> unp port-template auth-template mac-authentication
-> unp port-template auth-template 802.1x-authentication
-> unp port-template auth-template mac-authentication pass-alternate corporate
-> unp port-template auth-template 802.1x-authentication pass-alternate
corporate
```

11 Assign the port template to a UNP port.

```
-> unp port 2/1/1 port-template auth-template
```

12 Create a Captive Portal profile.

```
-> captive-portal-profile cp-profile
-> captive-portal-profile cp-profile aaa-profile ag-aaa-profile
```

13 Add a Captive Portal authentication pass policy list to the Captive Portal profile.

```
-> captive-portal-profile cp-profile authentication-pass policy-list cp-default-
list
```

14 Enable Captive Portal authentication for the UNP profile and assign the Captive Portal profile to that UNP profile.

```
-> unp profile guest captive-portal-authentication
-> unp profile guest captive-portal-profile cp-profile
```

How it Works

In this application example, traffic received on the UNP port triggers the following actions on the switch:

- Traffic from a supplicant device triggers the 802.1X authentication process.
- If 802.1X authentication passes, the client is classified into the UNP profile name returned from the RADIUS server or classified into the “corporate” UNP profile.
- If 802.1X authentication fails, the client is classified into the default UNP profile associated with the UNP port. This happens because rule classification is disabled on the UNP port. Captive Portal authentication is enabled for the default UNP profile.
- Traffic from a non-supplicant device triggers the MAC authentication process.
- If MAC authentication passes, the client is classified into the UNP profile name returned from the RADIUS server or classified into the “corporate” UNP profile.
- IF MAC authentication fails, the client is classified into the default UNP profile associated with the UNP port. This happens because rule classification is disabled on the port. Captive Portal authentication is enabled for the default UNP profile.

- The Captive Portal authentication pass condition applies a new access policy list to the client.
- If Captive Portal authentication fails, the client remains in a built-in Captive Portal pre-login state.

Application Example 5: IP Phone (LLDP Network Policy TLV/ Mobile Tag)

In this example, network access control is provided for the following IP phone devices:

- An IP phone enabled for LLDP Network Policy TLV and connected to a switch that is configured to send a Network Policy TLV with tagged VLAN.
- An IP phone that is statically configured to tag traffic with a specific VLAN.

The VLAN associated with the UNP profile to which the IP phone is assigned, must be tagged on the port after authentication. The following configuration steps provide a brief tutorial for how to achieve this:

1 Configure a RADIUS server.

```
-> aaa radius-server alu-authserver host 10.242.254.101 hash-key secret
retransmit 3 timeout 2 auth-port 1812 acct-port 1813
```

2 Create an AAA profile to pre-define and apply a specific AAA configuration for this example.

```
-> aaa profile ag-aaa-profile device-authentication 802.1x alu-authserver
-> aaa profile ag-aaa-profile accounting 802.1x alu-authserver
-> aaa profile ag-aaa-profile device-authentication mac alu-authserver
-> aaa profile ag-aaa-profile accounting mac alu-authserver
-> aaa profile ag-aaa-profile device-authentication captive-portal alu-
authserver
-> aaa profile ag-aaa-profile accounting captive-portal alu-authserver
```

3 Create the required VLANs.

```
-> vlan 10 admin-state disable name vlan-block
-> vlan 20 admin-state enable name vlan-corporate
-> vlan 30 admin-state enable name vlan-guest
-> vlan 40 admin-state enable name vlan-voice
```

4 Create the required UNP profiles.

```
-> unp profile corporate
-> unp profile guest
-> unp profile corporate-voice
```

5 Map each of the UNP profiles to an appropriate VLAN.

```
-> unp profile corporate map vlan 20
-> unp profile guest map vlan 30
-> unp profile corporate-voice map vlan 40
```

6 Enable mobile tagging on the UNP profile.

```
-> unp profile corporate-voice mobile-tag
```

7 Create a default UNP profile to assign to the UNP port.

```
-> unp profile def_unp
```

8 Map the default UNP profile to VLAN 10.

```
-> unp profile def_unp map vlan 10
```

9 Create a UNP port template to pre-define and apply configuration parameters to the UNP port.

```
-> unp port-template voice-template
```

10 Set the default profile parameter for the port template to “def_unp”.

```
-> unp port-template voice-template default-profile def_unp
```

11 Set the MAC and 802.1X authentication parameters to “enable” for the port template. Can also define a pass alternate UNP profile for the template in case the RADIUS server does not return a UNP profile name when 802.1X or MAC authentication passes.

```
-> unp port-template voice-template mac-authentication
-> unp port-template voice-template 802.1x-authentication
-> unp port-template voice-template mac-authentication pass-alternate corporate
-> unp port-template voice-template 802.1x-authentication pass-alternate profile
corporate
```

12 Assign the port template to a UNP port.

```
-> unp port 3/1/1-2 port-template voice-template
```

13 Enable LLDP IP Phone classification.

```
-> unp classification lldp med-endpoint ip-phone profile1 corporate-voice
```

14 Configure LLDP on the port.

```
-> lldp port 3/1/1-2 lldpdu tx-and-RX
-> lldp network-policy 1 application voice vlan 40 l2-priority 6
-> lldp port 3/1/1-2 med network-policy 1
```

How it Works

The expected traffic flow for this application example is as follows:

- EAP frames are the first frames sent by the IP phone on link up. The EAP frames are untagged.
- If the IP phone is a supplicant, 802.1X authentication is initiated. If the phone is a non-supplicant, MAC authentication is initiated.
- If the RADIUS server is configured to return the correct UNP profile name for the voice device, then that profile is applied when the device passes authentication.
- If the RAIDUS server is not configured to return the UNP profile name, then the 802.1X or MAC authentication pass alternate UNP profile is applied. Mobile tagging is enabled for the authentication pass alternate UNP profile.
- If 802.1X authentication fails, the device is blocked. If MAC authentication fails, the device must be enabled for LLDP IP phone classification.
- The VLAN assigned after the authentication and classification pass should be the same VLAN referred to in the configuration steps for this application example—the VLAN in the LLDP Network TLV advertisement and the VLAN associated with the UNP profile assigned to the IP phone. This VLAN should also be tagged on the UNP port, so that the traffic to or from the IP phone can be tagged.

- LLDP frames are exchanged between the IP phone and the switch. This traffic will be untagged but will be accepted by the switch since these are control frames.
- Subsequent data traffic will be tagged with the right VLAN after the LLDP exchange; this traffic will be accepted because the VLAN is a tagged member of the port.

Application Example 6: Restricted Role (Policy List) Assignment

This application example demonstrates post-authentication role assignment through the QMR feature, a location-based policy, and a time-based policy.

Quarantine Manager and Remediation (QMR)

A client MAC address is determined to be in a quarantined state when one of the following occurs:

- The OmniVista Quarantine Manager (OVQM) application receives a TRAP indicating that the MAC address has to be quarantined. The TRAP may come from a network anomaly detection application or from an intrusion detection system (IDS) running in the same subnet as the client.
- A list containing the quarantined MAC address is manually configured on OVQM.
- A list containing the quarantined MAC is manually configured on every switch in the network.

After the list of quarantined MAC addresses is known, OVQM can add these addresses to the Quarantine MAC group and push the configuration to the switches in a logical group or to all switches. Access Guardian then moves the users associated with the quarantined MAC addresses to a QMR restricted role.

There is a built-in policy list associated with the QMR restricted role that can be replaced with a user-defined policy list. For example, the administrator may want to use the following explicit policy list for QMR redirection instead of the built-in policy list:

```
-> policy service http80 destination tcp-port 80
-> policy service http443 destination tcp-port 443
-> policy service http8080 destination tcp-port 8080
-> policy service http8081 destination tcp-port 8081
-> policy service group alaRestrictedHttpSG http80 http443 http8080 http8081
-> policy condition qmr_traffic service group alaRestrictedHttpSG
-> policy action qmr_action redirect module qmr
-> policy rule qmr_rule condition qmr_traffic action qmr_action no default-list
-> policy list qmr_list type unp
-> policy list qmr_list rules qmr_rule
-> qos apply
```

With minor changes (such as changing the redirect module option to “captive-portal” or “byod”), this example policy list may also be useful for internal Captive Portal and OmniSwitch BYOD redirection.

The following OmniSwitch configuration demonstrates assigning a different role (explicit policy list) to a quarantine user as well as an example of configuring QMR on the switch:

1 Use the **unp restricted-role policy-list** command to assign a new policy list to replace the built-in QMR policy list. This is an optional command.

```
-> unp restricted-role qmr policy-list qmr_list
```

2 Configure the name of the Quarantine MAC Group. The default name of this group is “Quarantined”, so changing the name is optional. To change the name of this group, use the **qos quarantine mac-group** command.

```
-> qos quarantine mac-group bad-macs
```

Make sure the name of this group on the OmniSwitch matches the group name used by OVQM

3 The Quarantine MAC address group is populated from the same group located on an LDAP server. However, it is also possible to manually add MAC addresses to the MAC address group on the switch.

```
-> policy mac group Quarantined 00:9a:2d:00:00:10
```

4 Apply the QoS configuration for the MAC group name change (Step 2) and the manual MAC address changes (Step 3) to take effect on the switch.

```
-> qos apply
```

5 Add the IP address and subnet of the remediation server to a list of allowed IP addresses using the **qmr quarantine allowed-name** command. The allowed IP list specifies IP network addresses that a device is allowed to communicate with while in a quarantined state.

```
-> qmr quarantine allowed-name it-helpdesk 10.1.1.0 ip-mask 255.255.255.0
```

6 Create the path to the remediation server using the **qmr quarantine path** command.

```
-> qmr quarantine path www.remediate.com
```

7 If there is no quarantine path to redirect to, use the **qmr quarantine page** command to direct the switch to send a quarantine page to inform the user of the quarantined state.

```
-> qmr quarantine page enable
```

For more information about the QMR feature, see "Using Quarantine Manager and Remediation" on page 30-128.

UNP Profile - Time Policy

A time-based policy is associated with a UNP profile to define a validity period during which the profile applies a role (policy list) to the user. When a user classified into the UNP profile violates the validity period, the user is moved into an Unauthorized role.

There is a built-in policy-list associated with the Unauthorized role that can be replaced with a user-defined policy list. The following OmniSwitch configuration demonstrates assigning a different role to a user in an Unauthorized state as well as an example of configuring time based policies:

1 Create different validity periods as required. Different validity periods can be defined and assigned to different UNP profiles.

```
-> unp policy validity-period employee-shift-time days monday tuesday wednesday
thursday friday timezone PST hours 6:00 TO 18:00

-> unp policy validity-period guest-time days Monday tuesday wednesday thursday
friday saturday sunday timezone PST hours 9:00 TO 18:00
```

2 Assign the time policies created in Step 1 to an existing UNP profile.

```
-> unp profile UNP-employee period-policy employee-shift-time
-> unp profile UNP-guest period-policy guest-time
```

3 Assign a new policy list to replace the built-in policy list for the Unauthorized role.

```
-> unp restricted-role unauthorized policy-list unauthorized-time
```

UNP Profile - Location Policy

A location-based policy is associated with a UNP profile to define a specific location from which a device can access the network. When a user classified into the UNP profile violates the location policy, the user is moved into an Unauthorized role.

There is a built-in policy-list associated with the Unauthorized role that can be replaced with a user-defined policy list. The following OmniSwitch configuration demonstrates assigning a different role to a user in an Unauthorized state as well as an example of configuring time based policies:

1 Create different location policies as required. Different location policies can be defined and assigned to different UNP profiles.

```
-> unp policy validity-location employee-location port 1/1/1-24
-> unp policy validity-location guest-location port 1/1/15-24
```

2 Assign the location policies created in Step 1 to an existing UNP Edge profile.

```
-> unp profile UNP-employee location-policy employee-location
-> unp profile UNP-guest period-policy guest-location
```

3 Assign a new policy list to replace the built-in policy list for the Unauthorized role.

```
-> unp restricted-role unauthorized policy-list unauthorized-location.
```

Verifying Access Guardian Users

The following UNP **show** commands provide a centralized way to verify the status of users authenticated and classified through Access Guardian security mechanisms:

show unp user
show unp user status
show unp user details

This section provides sample display outputs from the **show unp user** commands. For more information about the display outputs for the other **show unp** commands, see "Verifying the Access Guardian Configuration" on page 30-149.

1 The **show unp user** command displays information about the MAC addresses learned on UNP ports and link aggregates:

```
-> show unp user
                                          User
Port  Username           Mac address      IP      Vlan Profile   Type    Status
-----+------------------+----------------+-------+----+---------+-------+-------
1/1/1 00:00:00:00:00:01  00:00:00:00:00:01  1.1.1.1  10  unp-1     Access  Active
1/1/2 00:00:00:00:00:02  00:00:00:00:00:02  1.1.1.2  11  unp-2     Bridge  Active
1/1/3 guest_user         00:00:00:00:00:04  1.1.1.4  20  unp-guest Access  Active
1/1/7 00:00:00:00:00:07  00:00:00:00:00:07  1.1.1.7  11  unp-emp   Bridge  Active
0/10  Employee-001       00:00:00:00:00:03  1.1.1.3  12  unp-emp   Bridge  Active
0/12  00:00:00:00:00:14  00:00:00:00:00:14  1.1.2.4  20  unp-7     Bridge  Active

Total users : 6

-> show unp user port 1/1/3
                                          User
Port   Username           Mac address      IP      Vlan  Profile     Auth   Role
------+------------------+----------------+-------+-----+-----------+------+------
1/1/3  guest_user         00:00:00:00:00:04 1.1.1.4  20   unp-guest   8021X  Guest

Total users : 1

-> show unp user linkagg 10
                                         User
Port   Username           Mac address     IP      Vlan  Profile     Auth   Role
------+-----------------+----------------+-------+-----+-----------+-----+--------
0/10   Employee-001      00:00:00:00:00:03 1.1.1.3  12   unp-emp     8021X Employee

Total users : 1

-> show unp user profile unp-emp
                                         User
Port   Username           Mac address     IP      Vlan  Profile     Auth   Role
------+-----------------+----------------+-------+-----+-----------+-----+--------
1/1/7  00:00:00:00:00:07 00:00:00:00:00:07 1.1.1.7  11   unp-emp     MAC   Employee
0/10   Employee-001      00:00:00:00:00:03 1.1.1.3  12   unp-emp     8021X Employee

Total users : 2
```

```
-> show unp user authentication-type mac
                                                User
Port   Username            Mac address       IP       Vlan  Profile     Auth   Role
------+-----------------+-----------------+-------+-----+-----------+-----+--------
1/1/7  00:00:00:00:00:07 00:00:00:00:00:07 1.1.1.7  11   unp-emp     MAC   Employee
0/12   00:00:00:00:00:14 00:00:00:00:00:14 1.1.2.4  20   unp-7       MAC   Employee

Total users : 2
```

2 The **show unp user status** command displays the status of the authentication and validation process for MAC addresses learned on a UNP port or link aggregate:

```
-> show unp user status port 1/1/1
                       Profile  Profile Authentication Role  Role                Restricted
Port  Mac address      Name     Source  Type   Status  Name  Source  CP  Redirect Access
-----+----------------+--------+-------+------+-------+-----+-------+---+--------+----------
1/1/1 00:00:00:00:00:05 Prf1     Radius  8021x  Passed  emp1  Profile  Y  -        -

Total users : 1

-> show unp user status linkagg 100
                       Profile  Profile Authentication Role  Role                Restricted
Port  Mac address      Name     Source  Type   Status  Name  Source  CP  Redirect Access
-----+----------------+--------+-------+------+-------+-----+-------+---+--------+----------
0/100 00:00:00:00:00:06 Prf3     Radius  8021x  Passed  emp1  Profile  Y  -        -

Total users : 1

-> show unp user status authentication type MAC
                       Profile  Profile Authentication Role  Role                Restricted
Port  Mac address      Name     Source  Type   Status  Name  Source  CP  Redirect Access
-----+----------------+--------+-------+------+-------+-----+-------+---+--------+----------
1/1/2 00:00:00:00:00:15 Prf2     Alt     MAC    Passed  emp2  Profile  Y  -        -

Total users : 1
```

3 The **show unp user details** command displays additional details about MAC addresses learned on a UNP port or link aggregate:

```
-> show unp user details port 1/1/10

Port:  1/1/10
  MAC-Address: 00:00:00:00:00:01
Sap                              : -,
Service ID                       : -,
VNID                             : -,
ISID                             : -,
Access Timestamp                 : 04/01/1970 18:45:26,
User Name                        : guest1,
IP-address                       : 10.0.0.1,
Vlan                             : 10,
Authentication Type              : 802.1X,
Authentication Status            : Authenticated,
Authentication Failure Reason    : -,
Authentication Retry Count       : -,
Authentication Server IP Used    = 10.135.62.129,
Authentication Server Used       = rad1,
Server Reply-Message             = -,
Profile                          : Employee,
```

```
Profile Source                  : RADIUS Server Profile,
Profile From Auth Server        : Employee,
Classification profile rule     : -,
Role                            : Employee,
Role Source                     : Profile,
User role rule                  : -,
Restricted Access               : No,
Location Policy Status          : Passed,
Time Policy Status              : Passed,
Captive-Portal Status           : -,
QMR Status                      : Passed,
Redirect Url                    : -,
SIP Call Type                   = Not in a call,
SIP Media Type                  = None,
Applications                    = None

  MAC-Address: 00:00:00:00:00:02
Sap                             : -,
Service ID                      : -,
VNID                            : -,
ISID                            : -,
Access Timestamp                : 06/01/1989 20:45:26,
User Name                       : guest2,
IP-address                      : 20.0.0.1,
Vlan                            : 20,
Authentication Type             : MAC,
Authentication Status           : Authenticated,
Authentication Failure Reason   : -,
Authentication Retry Count      : -,
Authentication Server IP Used   = 10.135.62.129,
Authentication Server Used      = rad1,
Server Reply-Message            = -,
Profile                         : Contractor,
Profile Source                  : RADIUS Server Profile,
Profile From Auth Server        : Contractor,
Classification profile rule     : -,
Role                            : Contractor,
Role Source                     : Profile,
User role rule                  : -,
Restricted Access               : No,
Location Policy Status          : Passed,
Time Policy Status              : Passed,
Captive-Portal Status           : Passed,
QMR Status                      : -,
Redirect Url                    : -,
SIP Call Type                   = Normal Call,
SIP Media Type                  = Video,
Applications                    = None

-> show unp user details linkagg 100
Port:  0/100
  MAC-Address: 00:00:00:00:00:03
Sap                             : -,
Service ID                      : -,
VNID                            : -,
ISID                            : -,
Access Timestamp                : 02/01/2013 20:45:26,
User Name                       : guest3,
IP-address                      : 30.0.0.1,
```

```
Vlan                            : 30,
Authentication Type             : MAC,
Authentication Status           : Authenticated,
Authentication Failure Reason   : -,
Authentication Retry Count      : -,
Authentication Server IP Used   = 10.135.62.129,
Authentication Server Used      = rad1,
Server Reply-Message            = -,
Profile                         : Contractor,
Profile Source                  : Auth - Pass - Default UNP,
Profile From Auth Server        : Employee [Not Configured],
Classification profile rule     : -,
Role                            : Contractor,
Role Source                     : Profile,
User role rule                  : -,
Restricted Access               : No,
Location Policy Status          : Passed,
Time Policy Status              : Passed,
Captive-Portal Status           : Passed,
QMR Status                      : -,
Redirect Url                    : -,
SIP Call Type                   = Not in a call,
SIP Media Type                  = None,
Applications                    = ;Facebook;rediff;
```

For more information about the displays that result from these commands, see the "Access Guardian Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Logging Users Out of the Network

In the event that it becomes necessary to manually log a user out of the network, the **unp user flush** command is available to the switch administrative user. The following parameters are available with this command to specify which user MAC addresses are flushed:

- **mac-address**—Flushes the user device with the specified source MAC address. For example:

  ```
  -> unp user flush mac-address 00:2a:95:00:3a:10
  ```

- **port** *chassis/slot/port1*[-*port2*]—Flushes the MAC addresses of all users connected to the specified switch port. For example:

  ```
  -> unp user flush port 1/9
  ```

- **linkagg** *agg_id*[-*agg_id2*]—Flushes the MAC addresses of all users connected to the specified link aggregate. For example:

  ```
  -> unp user flush linkagg 10
  ```

- **sap-id [linkagg]** *sap_id*—Flushes the MAC addresses of all users learned on a specific Service Access Point (SAP). A SAP ID is comprised of a device-facing port or link aggregate (referred to as a service access port) and an encapsulation value that is used to identify the type of device traffic to map to the service that is associated with the SAP. Use the optional **linkagg** parameter if the SAP ID is for a link aggregate. For example:

  ```
  -> unp user flush sap-id 1/1/2:50
  ```

- **service-id** *service_id*—Flushes the MAC addresses of all users learned on a specific service ID. For example:

```
-> unp user flush service-id 10
```

- **authentication-type {802.1x, mac, none}**—Flushes the MAC addresses of all users authenticated with the specified authentication type or users that have not been authenticated. For example:

```
-> unp user flush authentication-type 802.1x
-> unp user flush authentication-type mac
-> unp user flush authentication-type none
```

- **profile** *profile_name*—Flushes the MAC addresses of all users associated with the specified UNP profile name. Combine this parameter with the **mac-address** parameter to flush a specific user associated with the specified profile name. For example:

```
-> unp user flush profile EP-1
-> unp user flush profile EP-1 mac-address 00:da:95:11:22:01
```

Logging a group of users out of the network is particularly useful if configuration changes are required to any Access Guardian features. The **unp user flush** command is only available to the switch admin user. The admin account, however, is protected from any attempts to log out the admin user.

Verifying the Access Guardian Configuration

A summary of the **show** commands used for verifying the Access Guardian configuration is given here:

show unp global configuration	Displays the global UNP parameter settings for the switch
show unp port	Displays the UNP configuration for a port or link aggregate.
show unp port 802.1x statistics	Displays 802.1X statistics for a UNP port or link aggregate on which 802.1X authentication is enabled.
show unp port configured-vlans	Displays the VLANs assigned to UNP bridge ports or link aggregates.
show unp port-template	Displays the UNP port template configuration.
show unp domain	Displays the UNP domain ID configuration. This type of ID is used to group UNP ports into a logical domain.
show zeroconf server policy-instances	Displays the UNP profile configuration.
show unp profile map	Displays the VLAN ID or service mapping configuration assigned to a UNP profile.
show unp classification	Displays the UNP classification rule configuration for individual and binding rules.
show unp classification-rule	Displays the UNP extended classification rule configuration.
show unp user-role	Displays the UNP user-defined role configuration.
show unp restricted-role	Displays the explicit policy list configuration for built-in restricted roles.
show aaa device-authentication	Displays a list of RADIUS servers assigned to provide 802.1X, MAC, or Captive Portal authentication.
show aaa accounting	Displays a list of RADIUS servers assigned to provide 802.1X, MAC, or Captive Portal accounting.
show aaa config	Displays the AAA parameter configuration for 802.1X, MAC, and Captive Portal sessions
show aaa profile	Displays the AAA profile configuration.
show captive-portal configuration	Displays the global Captive Portal settings for the switch.
show captive-portal profile-names	Displays the Captive Portal configuration for the switch.
show qmr	Displays the global QMR settings for the switch.
show quarantine mac group	Displays the contents of the Quarantined MAC address group.

For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Bring Your Own Devices (BYOD) Overview

The OmniSwitch implementation of Bring Your Own Devices (BYOD) leverages the OmniVista Unified Policy Access Manager (UPAM) or the ClearPass Policy Manager (CPPM) and Access Guardian features on the OmniSwitch. BYOD can be implemented on a campus, branch offices, Internet edge, and converged access networks. It allows a wired guest, device, or authenticated user to connect to the network through an OmniSwitch edge device using UPAM or CPPM for unified authentication.

The BYOD support on the OmniSwitch provides the following:

- Unified access policy management solution for Wireline networks using UPAM or CPPM.
- Integration with Access Guardian UNP, 802.1X authentication, and MAC authentication.

Note: For additional information, refer to the following:

- "Access Guardian Overview" on page 30-12 for information about UNP device authentication and classification.
- OmniAccess WLAN documentation.
- OmniVista Unified Policy Access Manager documentation for in-depth OmniSwitch and server configuration requirements.
- ClearPass Policy Manager documentation for in-depth server configuration and licensing requirements.

- RADIUS Change of Authorization (CoA):
  - Provides a mechanism to change AAA attributes of a session after authentication.
  - Sends the New Profile as an attribute in the message.
  - Sends a Disconnect Message to terminate a user session and discard all user context.
- A validated BYOD solution using UPAM or CPPM with CoA and the OmniSwitch.
- Restricted access to the network and validation for end user devices, including employees with IT supplied devices, IP phones, employees personal devices, guest devices, access points, cameras, and silent devices (such as printers).
- UPAM or CPPM can act as a RADIUS server for new deployments or RADIUS proxy for existing networks.
- Captive Portal redirect using a dynamic redirect URL Vendor Specific Attribute (VSA).

Key Components of a BYOD Solution

The OmniSwitch BYOD solution comprises of the following main components:

- The network infrastructure consisting of both wireless and wireline networks. The OmniSwitch leverages the Access Guardian features, such as 802.1X (supplicant) and MAC (non-supplicant) authentication and classification through the Universal Network Profile (UNP) framework to support the BYOD solution.
- The UPAM and CPPM both interact with wireless and wireline networks acting as a RADIUS server or RADIUS server proxy. The UPAM and CPPM provide policy management, guest access, onboarding, and posture checking capabilities.

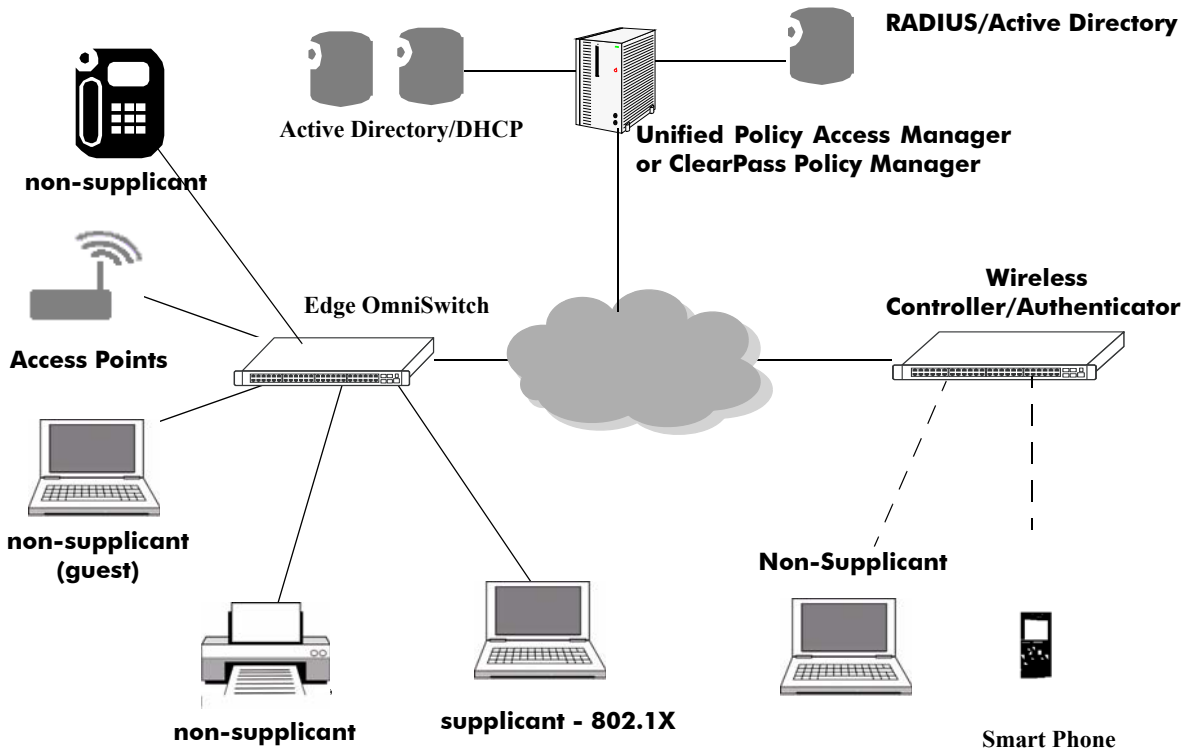


Figure 30-11 : BYOD Network Illustration

Unified Policy Access Manager and ClearPass Policy Manager

The OmniSwitch BYOD solution requires the association and configuration of the OmniVista Unified Policy Access Manager (UPAM) or the ClearPass Policy Manager (CPPM).

Note. This section describes the various services, features, and settings specific to CPPM. For information about the various UPAM services, features, and settings, refer to the OmniVista UPAM documentation.

ClearPass Guest

The BYOD solution supports guest self registration, sponsored guest access, and pre-registration of guest devices using MAC and Captive Portal authentication.

- Self Registration
  - An integrated external Captive Portal for guest or visitor registration.
  - Redirection to a customizable guest registration Captive Portal
- Sponsored Access
  - SMS and text email notification

ClearPass Policy Manager

ClearPass provides a user and device-independent framework that supports any BYOD initiative, large or small, by providing:

- Self-service onboarding, provisioning, and revocation of access for all major mobile devices.
- Device profiling as a basis for grooming traffic and improving network security based on device category, such as:
  - Device Category - Computer, Printer, AP
  - OS Family - MAC, Android, Windows, Linux
  - Device name and OS version
  - Useful for wired devices such as printers, access points, IP Phones, and cameras
- Controlled access and remediation for compromised devices
- Device disconnect if device signature changes
  - Secure guest network access with simplified workflows.

ClearPass Onboard

The BYOD solution supports the following services for device on-boarding and device management for guest and registered devices:

- Automatic configuration of Wireless, Wired 802.1X, VPN settings of personal and corporate devices that are connecting to the network for the first time.
- Management of digital certificates.
- Device on-boarding system is integrated with the external Captive Portal, which is separate from the internal OmniSwitch Captive Portal.
- Integration with the Enterprise Active Directory for authentication of employee credentials before device credentials are issued.
- Device provisioning supported through Aruba Quick Connect or Apple OTA API.

- Quick Connect supports native supplicants on Windows Vista, XP, 7, Apple, and Android devices.

ClearPass OnGuard

ClearPass OnGuard agents perform advanced endpoint posture checking to ensure compliance is met before the devices connect. The following functionalities are provided:

- Enhanced capabilities for endpoint compliance and control.
- Supports Microsoft, Apple, and Linux operating systems.
- Anti-virus, anti-spyware, firewall checks and more using the persistent or dissolvable agent.
- Optional auto-remediation and quarantine capabilities.
- System-wide endpoint messaging, notifications and session control.
- Centrally view the online status of all devices from the ClearPass Policy Manager platform.

OmniSwitch Integration with UPAM or CPPM for BYOD Support

Consider the following key points regarding OmniSwitch integration with UPAM or ClearPass for BYOD support:

- The same UNPs and access lists must be configured on both the OmniSwitch and UPAM or CPPM for proper alignment.
- The RADIUS server configuration on the OmniSwitch must point to the UPAM or CPPM in both proxy and server cases.
- A redirection server must be configured on the OmniSwitch that points to the UPAM or CPPM.
- Support for the Dynamic Vendor Specific Attribute (VSA) URL redirect is implemented using the OmniSwitch VSAs. The VSAs must be downloaded and installed on the ClearPass server; refer to the OmniVista UPAM documentation for information about how VSAs are installed on the UPAM server.
- A port bounce capability is configurable on the OmniSwitch to ensure a clean re-authentication process for non-supplicant devices.
- A PAUSE timer is configurable to flush out a user context (that is used for a welcome page or other user context information) on timer expiry.

RFC-3576 Attributes

RADIUS servers and the OmniSwitch can be configured with particular attributes defined in RFC 3576. These attributes carry specific authentication, authorization, and configuration details about RADIUS requests to and replies from the server. This section describes the attributes specific to an OmniSwitch BYOD solution.

Num.	CoA Attribute	Notes
40	**Disconnect-Request**	Disconnect Request sent by RADIUS/ClearPass server. • The Disconnect-Request RADIUS message contains the User-Name or the **Calling-Station-ID** attribute. • When the message contains both the User-Name and **Calling-Station-ID**, the MAC address is identified based on the **Calling-Station-ID** only.

41 DM-ACK	On reception of Disconnect request message (DM), all device authentication is removed from the switch. Disconnect request message (DM) Acknowledgment for RADIUS/UPAM or ClearPass authentication
42 DM-NACK	Disconnect request message (DM) Not Acknowledged
43 CoA-Request	CoA message is sent from UPAM or ClearPass Server. CoA-Request packets contain information for dynamically changing session authorizations.The following attributes are used: • **The User-Name:** AOS retrieves the MAC address associated to this user • **The Calling-Station-ID:** This explicitly specify the user MAC address When the message contains both the User-Name and **Calling-Station-ID**, the MAC address is identified based on the **Calling-Station-ID** only.
44 CoA-ACK	Supports a Change of Authorization-Request (CoA) message for RADIUS authentication.COA-ACK is sent by OmniSwitch to UPAM or ClearPass that has attributes MD5 hash value and Identifier.
45 CoA-NACK	COA-NACK message is sent from OmniSwitch. For NAK message, the Error-Cause attribute must be supported and filled accordingly.
Error-Cause	Supported as part of CoA-NAK and DM-NAK message. **Error-Cause** Scenarios: **Missing Attribute** - If User name and Calling station ID Filter ID not present **Invalid Request** - If Client context does not exist **Unsupported Attribute** - Request contains an unsupported Vendor-Specific attribute **Unsupported Service** - Request contains an unsupported or invalid service in Service-Type attribute **Nas Identification Mismatch** - Request contains one or more NAS identification attributes that does not match the identity of the NAS receiving the request **Administratively Prohibited** - NAS prohibiting the Request messages for the specified session **Session Context Not Found** - Session context identified in the request does not exist on the NAS **Resources Unavailable** - Request could not be honored due to lack of available NAS resources

Vendor-Specific Attributes for UPAM or ClearPass

The OmniSwitch RADIUS client supports attribute 26, which includes a vendor ID and some additional sub-attributes called subtypes. The vendor ID and the subtypes collectively are called Vendor Specific Attributes (VSAs).

For UPAM or ClearPass integration, the VSA dictionary must be updated with the "**Alcatel-Redirect-URL**" and the "**Alcatel-Access-Policy-List**" VSA that can be imported into the UPAM or ClearPass server. The following VSAs can be imported to the UPAM or ClearPass server:

Num.	ClearPass/RADIUS VSA	Type	Description
6	**Alcatel-Lucent-Port-Desc**	**string**	Description of the port. This attribute is currently defined in the Alcatel dictionary as: RADIUS attribute type = 26 (VSA) VSA Vendor ID = 800 VSA Type = 26 VSA format = string This attribute is included in all RADIUS messages sent by the OmniSwitch (Access-Request, Accounting-Request Start, Accounting-Request Interim and Accounting-Request Stop).The attribute is set with the alias configured for the port. When the alias is not set, VSA will be an empty string.
100	**Alcatel-Access-Policy-List**	**string**	Configures UPAM or ClearPass to the policy list associated with the UNP.
101	**Alcatel-Redirect-URL**	**string**	Configures UPAM or ClearPass to send redirection URL as part of RADIUS response redirecting the user Web traffic.

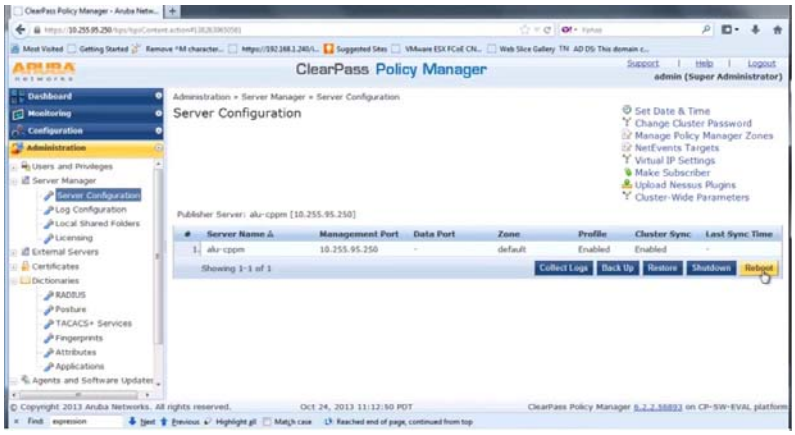
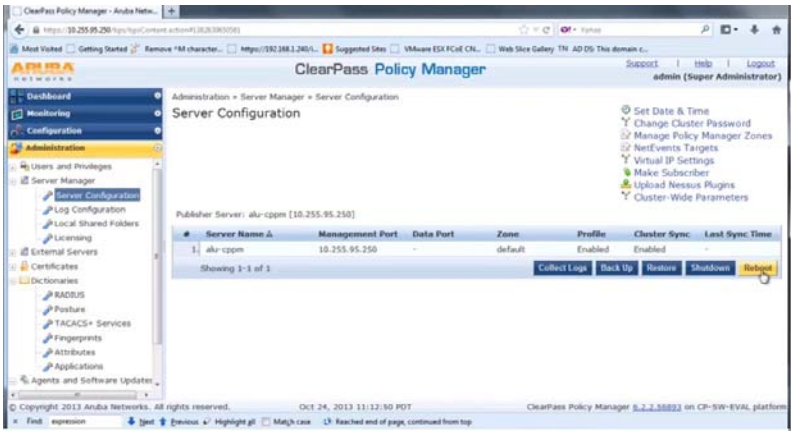
Importing the Alcatel-Lucent Enterprise Dictionary into CPPM

Note. The procedure described in this section is specific to importing the VSA dictionary into the CPPM server. For information about the VSA dictionary integration with UPAM, refer to the OmniVista UPAM documentation.

Perform the following to import the VSA dictionary into the CPPM server:

1. Download the **Alcatel-Lucent-Enterprise.xml** file from the Service & Support website.
2. Click on **Dictionary->Import Dictionary** and browse for the ***Alcatel-Lucent-Enterprise.xml*** file.
3. Click on **Server Configuration->Reboot** to reboot the server.

Figure 30-12 : Importing the Alcatel-Lucent Enterprise dictionary into CPPM

Import Dictionary: **Dictionary->Import Dictionary**	
Reboot CPPM **Server Configuration->Reboot**	

Port Bounce

A port bounce is used to terminate a user session and discard all associated session context for non-supplicants. This is done by disabling and re-enabling the port and clearing any authentication state for the devices on the port. A port bounce action is configurable through the **unp redirect port-bounce** command.

- Port bounce is used for MAC authenticated non-supplicant users.
- On receipt of a Disconnect Request (DM) or Change of Authorization (CoA) message, the OmniSwitch determines if the user needs to move or change VLANs.
- If the new UNP specifies a different VLAN ID, the port bouncing feature is enforced as per configuration for non-supplicants.
- When a device changes VLANs and it is the only device on the port, the switch port is bounced to ensure a clean reconnection and get the correct IP address through DHCP.
- Port bouncing is enforced only if the non-supplicant user is the only user on the port. Also if a CoA message is received for a non-supplicant user and port bouncing is disabled globally but is enabled on the port on which the non-supplicant user has been classified, then the port is bounced.

Pause Timer

The switch clears all authentication states of the device by pausing for a period of time. The value for this period of time is configurable through the **unp redirect pause-timer** command.

When non-supplicant devices are detected, the switch must pause for some period of time before redirection to the specified URLs. The pause mechanism is enforced when the following conditions are met.

- COA message received by the switch indicates VLAN movement for the non-supplicant user, and
- Port bouncing is disabled for the user port or disabled globally for the switch.

The pause mechanism ensures that all traffic from the user is dropped until the global pause timer expires and the corresponding user context is removed.

Note. The Port Bounce and Pause Timer functions apply only to non-supplicant devices. For supplicant devices, there is no difference whether Port Bounce is enabled or Pause Timer is enabled. The user context for supplicant devices is removed by triggering the re-authentication of the supplicant user and the device moves into a new UNP profile and VLAN after re-authentication.

Configuring OmniSwitch BYOD Support

BYOD is supported on UNP ports for supplicant and non-supplicant registered and guest users and devices.The BYOD solution leverages the existing Access Guardian UNP capability and is applicable only on UNP ports. The following general configuration tasks are required to ensure the necessary interaction between an OmniSwitch and the UPAM or CPPM server:

Note. Configure the OmniSwitch to interact only with the OmniVista UPAM server or the CPPM server.

- Configure the UPAM or CPPM server as an AAA RADIUS server.
- Set the switch to use the UPAM or CPPM server for 802.1X and MAC authentication. The authentication process will determine the UNP profile to which BYOD users are classified.
- Configure the UNP profiles that will be returned from the UPAM or CPPM server. Make sure the Captive Portal authentication flag is disabled on each of these profiles to ensure BYOD redirection.
- Configure the UPAM or CPPM server as the redirect server for the switch.
- Configure UNP port-based functionality on the switch ports that will connect to the user devices.
- Configure the OmniSwitch to relay DHCP traffic to the UPAM or CPPM server as well as the DHCP server, which assigns the IP addresses to the clients connected to the switch. UPAM or CPPM uses this information to assist with device profiling.
- Configure the UPAM or CPPM server with the IP address of the OmniSwitch. In addition, configure the UPAM or CPPM with the same shared secret that was assigned through the AAA RADIUS server configuration on the OmniSwitch.
- Configure the UPAM or CPPM server with the required services (for example, MAC authentication, 802.1X, and any generic RADIUS enforcement service) to support the following features.
  - Device profiling
  - Device Onboarding
  - Guest Registration
  - Posture check
  - Captive portal

The following generic configuration examples apply only to the OmniSwitch components for interaction with a UPAM or CPPM server. For more detailed application examples, refer to "BYOD Application Examples" on page 30-180.

Configuring the UPAM or CPPM server as an AAA RADIUS Server

The UPAM or CPPM server must be configured on the OmniSwitch as an AAA RADIUS server that will handle 802.1X and MAC authentication requests from the switch. Optionally, the OmniSwitch can also be set to use the UPAM or CPPM server for 802.1X and MAC accounting sessions as well. For example:

```
-> aaa radius-server cppm host 192.168.1.244 key e47ac0f11e9fa869 retransmit 3
timeout 2 auth-port 1812 acct-port 1813
-> aaa device-authentication 802.1x cppm
-> aaa device-authentication mac cppm
-> aaa accounting 802.1x cppm
-> aaa accounting mac cppm
```

Configuring UNP Profiles

Users connected to UNP-enabled ports are moved into a specific UNP profile based on the outcome of the authentication process. This type of profile is created using the **unp profile** command. For example:

```
-> unp profile UNP-guest
-> unp profile UNP-restricted
```

To support interaction with the UPAM or CPPM server, the same UNP profile name must be configured on both the OmniSwitch and on the UPAM or CPPM server. In addition, the Captive Portal authentication flag for the OmniSwitch profile must be disabled. For example:

```
-> no unp profile UNP-guest captive-portal-authentication
-> no unp profile UNP-restricted captive-portal-authentication
```

Once a UNP profile is created with the Captive Portal authentication flag disabled, then the profile must be mapped to a VLAN ID. Users classified into the profile are dynamically assigned to the associated VLAN ID. To assign a VLAN ID to a profile, use the **unp profile map vlan** command. For example:

```
-> unp profile UNP-guest map vlan 100
-> unp profile UNP-restricted map vlan 455
```

Configuring Redirection with Dynamic URLs

The redirect server and the URL returned by the server are used to present guest users with different web pages depending on what state of authentication they are in. HTTP traffic from the user is redirected towards the URL returned by the server. Use the **unp redirect-server** command to specify the IP address of the redirect server, which should match the IP address in the returned URL. For example:

```
-> unp redirect server ip-address 192.168.1.244
```

If the OmniSwitch redirect server IP address does not match the redirect IP address in the UPAM or CPPM server configuration, HTTP traffic is not redirected to the URL.

To allow the user device to access one other IP subnet, use the **unp redirect allowed-name** command. For example:

```
-> unp redirect allowed-name server2 ip-address 10.0.0.20 ip-mask 255.0.0.0
```

Configuring a Custom Redirect Policy

When UPAM or CPPM returns a UNP with a redirect URL VSA but without an Alcatel-Access-Policy-List VSA, the OmniSwitch applies a built-in QoS policy list to the user. The built-in list allows DNS, ICMP, ARP, DHCP, and redirects Web traffic to the configured redirect UPAM or CPPM server. However, the administrator may want to apply a custom QoS redirect policy list that will override the built-in policy.

To override the built-in list policy list with a custom policy list for BYOD redirection:

- Create a custom redirect policy list on the OmniSwitch. Make sure the list rules contain the following required items:
  - A QoS service group named “alaRestrictedHttpSG”.
  - A redirect module policy action with the **byod** option.
- Configure UPAM or CPPM to return the OmniSwitch list name in the Alcatel-Access-Policy-List VSA. The policy list name in the VSA must match the name of the custom redirect policy list.

The following is an example of a custom QoS redirect policy list:

```
-> policy service http80 destination tcp-port 80
-> policy service http8080 destination tcp-port 8080
-> policy service https443 destination tcp-port 443
-> policy service group alaRestrictedHttpSG http80 http8080 https443
-> policy port group pg1 1/1/1-20
-> policy condition byod service group alaRestrictedHttpSG
-> policy condition cppm source port group pg1 destination ip 135.254.163.143
-> policy action byod_action redirect module BYOD
-> policy action cppm
-> policy rule byod_rule condition byod action byod_action no default-list
-> policy rule cppm condition cppm action cppm no default-list
-> policy list reg_policy_list type unp
-> policy list reg_policy_list rules byod_rule cppm
-> qos apply
```

In this example, the custom QoS redirect policy list named “req_policy_list” is created with the required items (highlighted in blue). To allow this custom policy list to override the built-in policy, the UPAM or CPPM is configured to return the “req_policy_list” list name in the Alcatel-Access-Policy-List VSA.

Configuring UNP Port Authentication

UNP functionality and authentication settings must be enabled on the switch ports for the authentication process to begin. Use the **unp** configuration commands to enable UNP functionality on a port and specify the type of authentication to apply to traffic received on that port. For example:

```
-> unp port 1/1/4 port-type bridge
-> unp port 1/1/4 802.1x-authentication
-> unp port 1/1/4 mac-authentication
-> unp port 1/1/4 802.1x-authentication failure-policy mac
```

In this example, both 802.1X and MAC authentication is enabled on UNP port 1/1/4. In addition, an 802.1X authentication failure policy is configured for the port to direct the switch to attempt MAC authentication after a device on port 1/1/4 fails 802.1X authentication. This is particularly helpful when a guest device with built-in 802.1X credentials fails the initial 802.1X authentication process.

Configuring Port Bounce

Port bouncing is used to force a re-authentication for non-supplicant devices. By default, the port bounce action is enabled on all ports. Use the **unp redirect port-bounce** command to change the port bounce status. For example:

```
-> unp port 1/1/4 redirect port-bounce
-> unp port 1/1/4 redirect port-bounce
```

If a port is not specified with the **unp redirect port-bounce** command, the status is changed on a global basis for all UNP ports. For example:

```
-> unp redirect port-bounce enable
```

The port-level setting overrides the global setting for the port bounce operation.

Configuring the Pause Timer

The pause timer specifies an amount of time during which traffic from non-supplicant devices is filtered. By default, the pause time is set to zero. Use the **unp redirect pause-timer** command to set the pause timer value, in seconds. For example:

```
-> unp redirect pause-timer 120
```

BYOD Authentication Process Overview

This section describes the basic BYOD process with respect to the OmniSwitch interaction with the UPAM or ClearPass server.

Authentication for Registered Devices (802.1X)

The BYOD solution provides the following authentication process for registered devices (for example, IT issued employee devices):

1 When 802.1X authentication is enabled on a UNP port and the OmniSwitch detects a user device on that port, the authentication process is triggered to classify the user.

2 The OmniSwitch sends a request to the UPAM or ClearPass server that authenticates the user based on user credentials and the profiles and policies configured on the UPAM or ClearPass server.

3 UPAM or ClearPass classifies the user to a registered UNP and returns the UNP information to the OmniSwitch.

4 The OmniSwitch assigns the user to the UNP obtained from the UPAM or ClearPass server.

Authentication for Network Devices (MAC Authentication)

The BYOD solution provides the following MAC authentication process for network devices such as IP phones, printers, or access points.

1 When MAC authentication is enabled on a UNP port and the OmniSwitch detects a device on that port, the MAC authentication process is triggered to classify the device.

2 The OmniSwitch sends a request to the UPAM or ClearPass server that authenticates the device based on the device MAC address and the profiles and policies configured on the UPAM or ClearPass server.

3 UPAM or ClearPass classifies the device to a UNP and returns the UNP information to the OmniSwitch.

4 The OmniSwitch assigns the device to the UNP obtained from the UPAM or ClearPass server.

Authentication for Guest Devices and Employee Onboarding

The BYOD solution provides the following authentication process for guest devices and employee personal devices:

1 When MAC authentication is enabled on a UNP port and the OmniSwitch detects a device on that port, the MAC authentication process is triggered to classify the device.

2 UPAM or ClearPass initially classifies the device into a temporary UNP and returns a redirection URL that allows for guest registration or employee onboarding.

3 The OmniSwitch assigns the user to the temporary UNP name returned from UPAM or CPPM. Since redirection is also set, all DHCP or DNS traffic is allowed but HTTP traffic from the user is redirected towards the URL returned with the UNP.

4 The user is presented with a guest login page or an onboarding page to enter user credentials.

5 UPAM or ClearPass determines the appropriate role of the user after performing registration and sends the final UNP to the OmniSwitch through a RADIUS CoA request or through a RADIUS Access-Accept packet for onboarding.

Multicast Domain Name System

The Multicast Domain Name System (mDNS) is a resolution service that is used to discover services on a LAN. Using mDNS allows the resolution of host names to IP addresses within small networks without the need for a conventional DNS server. The mDNS protocol uses IP multicast User Datagram Protocol (UDP) packets and is implemented by Apple Bonjour, Avahi (LGPL), and Linux NSS-mDNS. In a BYOD network, mDNS is leveraged by providing wireless guests and visitors access to network devices, such as printers.

To resolve a host name, the mDNS client broadcasts a query message asking the host having that name to identify itself. The target machine then multicasts a message that includes its IP address. All machines in that subnet will use that information to update their mDNS caches.

For example, the Apple's Bonjour architecture implements the following three fundamental operations to support zero configuration networking service:

- Publication (Advertising a service)
- Discovery (Browsing for available services)
- Resolution (Translating service instance names to address and port numbers for use)

mDNS Work Flow

The following diagram represents an mDNS work flow setup. The wireless clients connected to Access point 1 (AP1) or Access Point 2 (AP2) request for the mDNS service offered

.

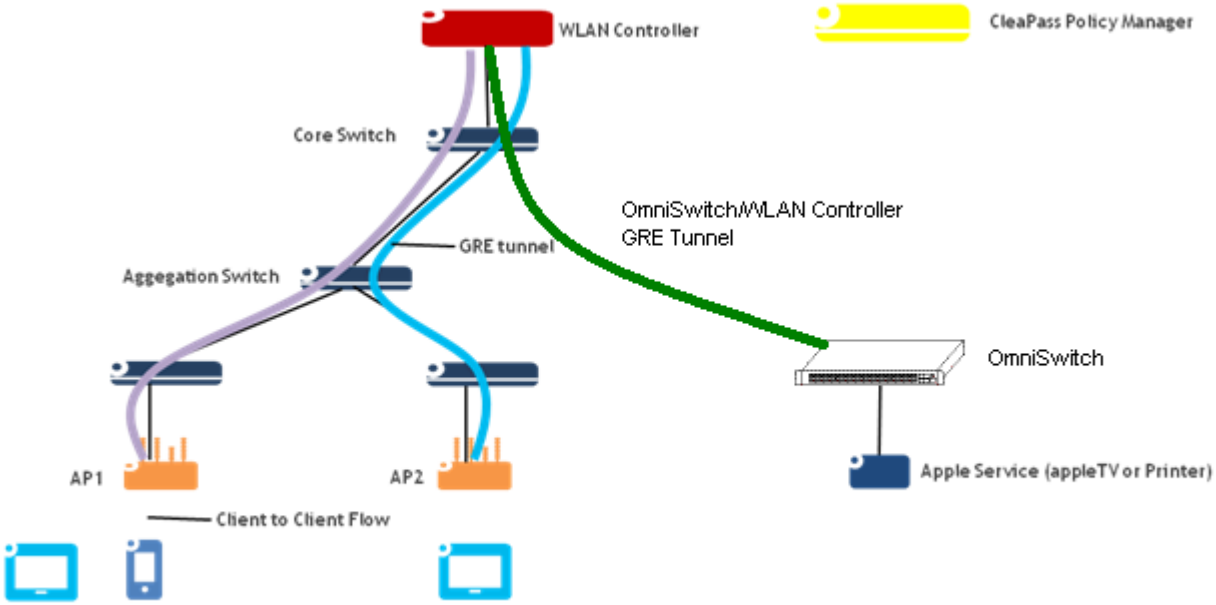


Figure 30-13 : mDNS Work Flow

The mDNS feature is enabled on the OmniSwitch to support the mDNS service. A Layer 2 GRE tunnel interface is configured from the WLAN controller to the OmniSwitch to relay the mDNS messages.

The mDNS message from the Bonjour capable wired service device is encapsulated and relayed from the OmniSwitch to the configured WLAN controller over the GRE tunnel. The WLAN controller then relays the mDNS messages received via the OmniSwitch GRE tunnel to the APs over the AP GRE tunnels.

Note. The WLAN controller uses a multicast optimization algorithm and forwards Bonjour response messages to targeted user devices, instead of all devices on all APs. This limits the unnecessary flooding of the Bonjour/mDNS traffic to improve the Wi-Fi performance.

Simple Service Discovery Protocol

The Digital Living Network Alliance (DLNA) is a standards organization that defines the guidelines for multimedia devices. It also certifies communication between devices allowing them to discover and recognize each other and share digital content. DLNA uses Universal Plug and Play (UPnP) for media management, discovery, and control. DLNA/UPNP uses the Simple Service Discovery Protocol (SSDP) to discover services, similar to Bonjour using mDNS for the same purpose. In the AirGroup solution, the WLAN controller acts as Bonjour and DLNA gateways allowing Layer 2 discovery protocols, such as mDNS and SSDP, to extend across Layer 3 boundaries through the gateway.

Along the lines of zero network configuration already supported by the OmniSwitch with mDNS, support of SSDP Relay for DLNA/UPnP enables the OmniSwitch to allow non-Apple devices to also discover services with minimal configuration by the administrator.

- DLNA/UPnP uses SSDP for dynamic discovery of services.
- The WLAN controller AirGroup feature has support for DLNA and acts as a DLNA controller, in addition to the support for mDNS.
- Similar to the OmniSwitch implementation of mDNS, the OmniSwitch relays SSDP packets to the WLAN controller through a Layer 2 GRE tunnel.

How SSDP Relay Works

All the SSDP packets coming in on an OmniSwitch are intercepted and tunneled through a GRE tunnel to the WLAN controller (acting as a gateway). The GRE tunnel is setup between the switch and the WLAN controller to tunnel both mDNS and SSDP frames. Similarly, traffic towards the SSDP clients/servers are sent back from the WLAN controller to the switch through the GRE tunnel. The reverse traffic is also intercepted and then sent unicast or multicast from the switch to the respective ports.

Messages Received by the OmniSwitch from Wired SSDP Devices

SSDP messages coming from wired SSDP service devices are relayed from the OmniSwitch to the WLAN controller using the associated GRE tunnel interface. The WLAN controller only supports Layer 2 GRE for SSDP, so frames sent from the OmniSwitch are encapsulated as follows:

DA-MAC	**SA - MAC**	**IP Header**	**Payload**
Gateway MAC to reach WLAN controller/ WLAN controller MAC	OmniSwitch MAC	Src IP: OmniSwitch IP Dst IP: WLAN controller IP IP protocol: GRE(47) GRE VER 00 00 GRE TYPE 00 00 (indicating L2 frame in the payload)	SSDP frame from the wired service/client (appended with 802.1q tag)

Messages Received by the OmniSwitch from the WLAN Controller

SSDP messages coming from the WLAN controller through the Layer 2 GRE tunnel interface are relayed towards the client device. When the OmniSwitch receives the encapsulated packets:

- The original SSDP frame is extracted from the GRE packet with the Layer 2 header.
- The switch obtains the VLAN ID from inside the 802.1Q header of the SSDP frame and floods the frame on that VLAN (untagged or tagged based on the egress port frame) towards the client.

SSDP frames received from the WLAN controller are encapsulated as follows:

DA-MAC	**SA - MAC**	**IP Header**	**Payload**
OmniSwitch MAC	WLAN controller MAC/intermediate router MAC	Src IP: WLAN controller IP Dst IP: OmniSwitch IP IP protocol: GRE GRE VER 00 00 GRE TYPE 00 00 (indicating L2 frame in the payload)	SSDP frame destined to the wired service/ client (appended with 802.1Q tag)

Configuration Guidelines

Consider the following guidelines when configuring SSDP Relay functionality for the switch:

- Configure the GRE tunnel interface before attempting to associate the interface with the SSDP tunnel relay. An IP address is required to bring the interface up; if necessary, specify a dummy IP address when configuring the interface.
- The GRE tunnel must also be configured on the WLAN controller. Refer to the OmniAccess WLAN user guide for additional information on configuring a WLAN controller.
- The SSDP and mDNS relay features both require an association with a GRE tunnel IP interface. If both of these features are required on the switch, configure both to use the same tunnel interface. There is only one Layer 2 GRE tunnel established between the switch and the WLAN controller, so both features need to use the same GRE tunnel.
- Only a Layer 2 IPv4 GRE tunnel is supported. However, IPv6 SSDP frames from the client are also relayed through the IPv4 GRE tunnel, as the payload includes the Layer 2 headers (only Layer 2 GRE mode is supported by the WLAN controller) regardless of the inner IPv6 header.

Application Example 1: Wired DLNA-Capable Client

In this application example, an active wired DLNA-capable client is attempting to print to a DLNA printer and stream to a DLNA TV. All the devices are connected to the OmniSwitch 6860, but the SSDP discovery phase has to flow through the WLAN controller. As a result, the Layer 2 GRE tunnel is required

to handle the initial contact between the devices. Once communication is established, subsequent traffic flows directly between the devices without going through the controller.

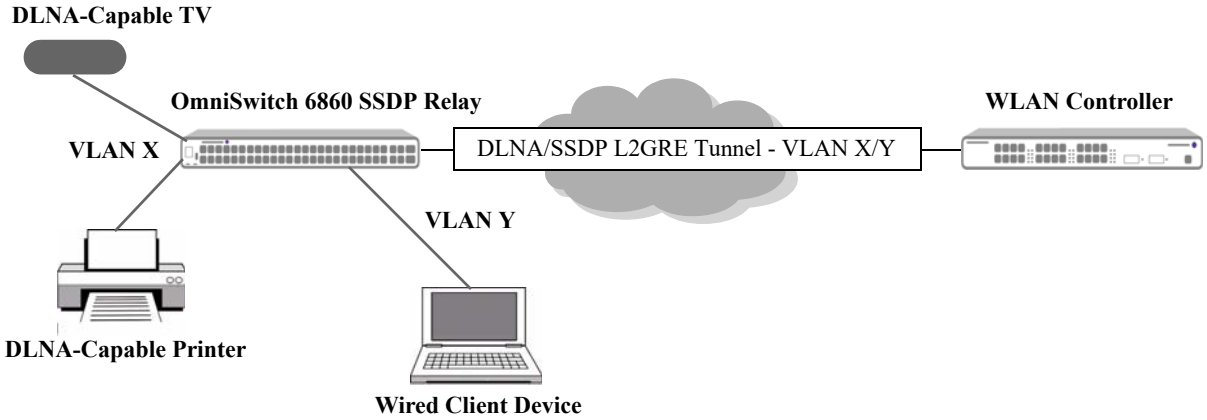


Figure 30-14 : SSDP - Wired DLNA-Capable Client

- The WLAN controller handles the SSDP discovery process between the wired client and end service devices. Encapsulated SSDP packets are sent between the controller and the OmniSwitch 6860 through the Layer 2 GRE tunnel until discovery is complete.
- The OmniSwitch 6860 relays SSDP packets from the DLNA-capable printer and TV to the WLAN controller through the Layer 2 GRE tunnel. See "Messages Received by the OmniSwitch from Wired SSDP Devices" on page 30-163.
- The OmniSwitch 6860 receives encapsulated SSDP packets through the GRE tunnel from the WLAN controller. The switch then extracts the SSDP frames from the encapsulated packets and forwards them to the destined end service device (DLNA-capable printer or TV). See "Messages Received by the OmniSwitch from the WLAN Controller" on page 30-164.

Application Example 2: Wireless DLNA-Capable Clients

In this application example, wireless DLNA-capable clients are accessing DLNA-capable devices with ClearPass Policy Manager (CPPM) integration for role and location information. All traffic between the wireless clients and the wired devices flows through the WLAN controller.

The CPPM integration is only available for wireless devices. If there are both wired and wireless DLNA-capable clients, the CPPM integration mode cannot be enabled on the WLAN controller. CPPM

integration is enabled at the global level; there is no independent control to apply this function only to wireless devices.

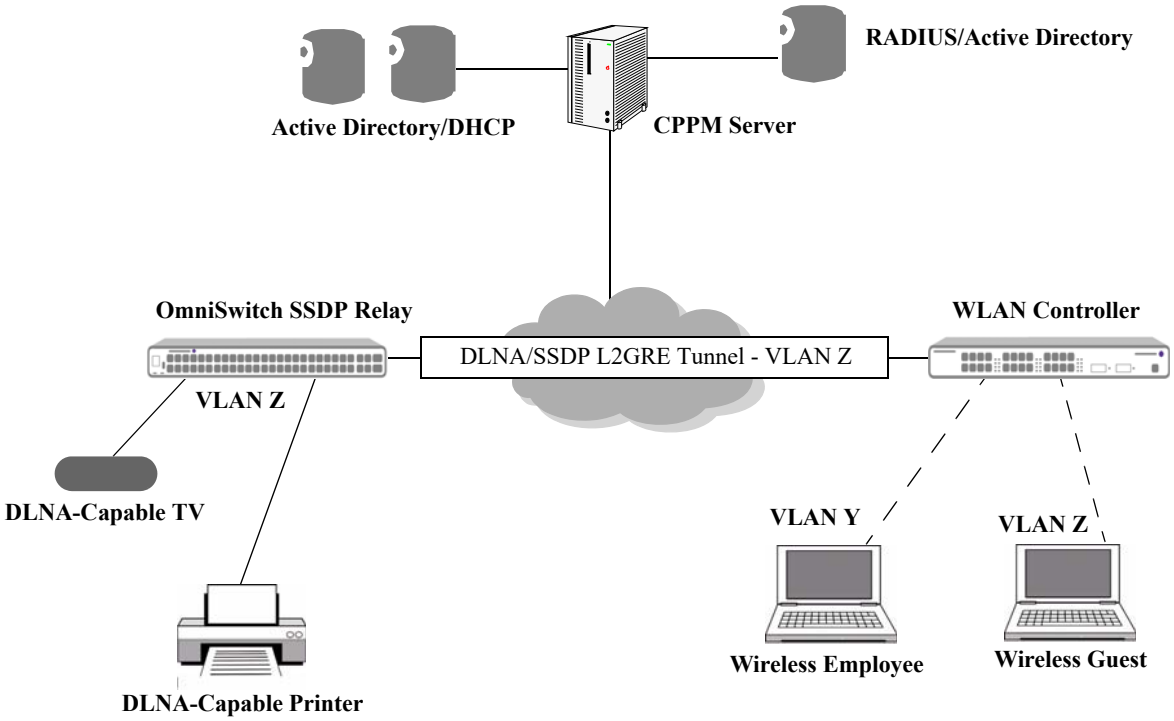


Figure 30-15 : SSDP - Wireless DLNA-Capable Clients

- The OmniSwitch 6860 relays SSDP packets from the DLNA-capable printer and TV to the WLAN controller through the Layer 2 GRE tunnel. See "Messages Received by the OmniSwitch from Wired SSDP Devices" on page 30-163.
- The WLAN controller sends a RADIUS request to the CPPM to request the policies needed to determine the end service devices that are available to the wireless clients. The following policies are the type of policies that may be returned:
  - Identity based policies.
  - Role based policies.
  - Location based policies.
  - Time based policies.
- The WLAN controller applies the service policies and sends encapsulated SSDP packets through the Layer 2 GRE tunnel to the end service devices connected to the OmniSwitch 6860.
- The OmniSwitch 6860 receives encapsulated SSDP packets through the GRE tunnel from the WLAN controller. The switch then extracts the SSDP frames from the encapsulated packets and forwards them to the destined end service device (DLNA-capable printer or TV). See "Messages Received by the OmniSwitch from the WLAN Controller" on page 30-164.

Zero Configuration Networking (mDNS and SSDP)

Zero configuration networking is a set of protocols that can be used to discover services. It allows communications between network devices and allowing them to advertise and share each others' resources. To resolve the service information, the mDNS client broadcasts a query message asking for the device containing the requested service to identify itself. The target machine then multicasts a message that includes its IP address and capabilities. All machines in that subnet will use that information to update their mDNS caches.

Bonjour is Apple's implementation of zero-configuration networking. Apple's Bonjour protocol, built on multicast DNS, is a Layer 2 non-routable protocol. This means that only clients on the same subnet as the AirPrint and AirPlay enabled devices can see those services. On a network that has multiple subnets, the multicast DNS advertisements will not reach users on different subnets. Enterprises, schools, universities and many other environments are typically built with multiple subnets, which means that although Apple services may be available to users, they will not be visible to them.

Similarly, DLNA (Digital Living Network Alliance) is a standard that is derived from UPnP (Universal Plug and Play). DLNA uses SSDP (Simple Service Discovery Protocol) for service discovery on the network. It provides the ability to share digital media services for Android or Windows devices. The SSDP protocol also has the same limitation of local subnet scope.

Hence, the zero configuration mDNS and SSDP solution is developed to extend mDNS and SSDP across Layer 3.

The zero configuration mDNS and SSDP solution allows:

- mDNS and SSDP compatible devices to discover network services across IP subnet boundaries.
- Selective sharing of network services based on sharing rules for mDNS or SSDP capable devices. Sharing rules are defined based on VLAN, access role profile (UNP), location, username and MAC address.
- A unified solution across a wired and wireless (Aruba AP) network.
- Multicast optimization over the wireless (Aruba AP) network.

The mDNS or SSDP packet handling across Layer 3 supports the following modes of operation:

- **Tunnel (Aruba) Mode**: Supports mDNS or SSDP compatible devices with Aruba controller with GRE tunnel protocol type 0x0. This is the default mode of operation.
- **Tunnel Standard Mode**: Supports tunneling for mDNS or SSDP compatible devices to an OmniSwitch responder with GRE tunnel protocol type 0x6558. Only mDNS or SSDP over IPv4 is supported.
- **Gateway Mode**: Supports mDNS or SSDP compatible devices to discover network services across IP subnet boundaries or VLANs. Only mDNS or SSDP over IPv4 is supported.
- **Responder Mode**: Supports mDNS or SSDP compatible devices with an OmniSwitch as a core switch (Responder). Only mDNS or SSDP over IPv4 is supported.

How It Works

Aruba Mode

Aruba Mode is configured if the network has only an Aruba wireless controller. In this mode, all the switches must be mDNS and SSDP enabled. All the edge switches must be configured to use the L2GRE tunnel of the Aruba wireless controller.

All the wireless traffic from Aruba APs is tunneled to the Aruba WLAN controller directly through the tunnel established between APs and Aruba WLAN controller.

The mDNS and SSDP traffic entering the edge switch is tunneled to the Aruba controller. The mDNS and SSDP traffic received back from the controller on the L2GRE tunnel is verified. If the packet is unicast, it is forwarded based on the destination. If the packet is multicast, it is forwarded to the VLAN based on the VLAN ID tag in the packet.

The following diagram represents a sample Aruba mode setup:

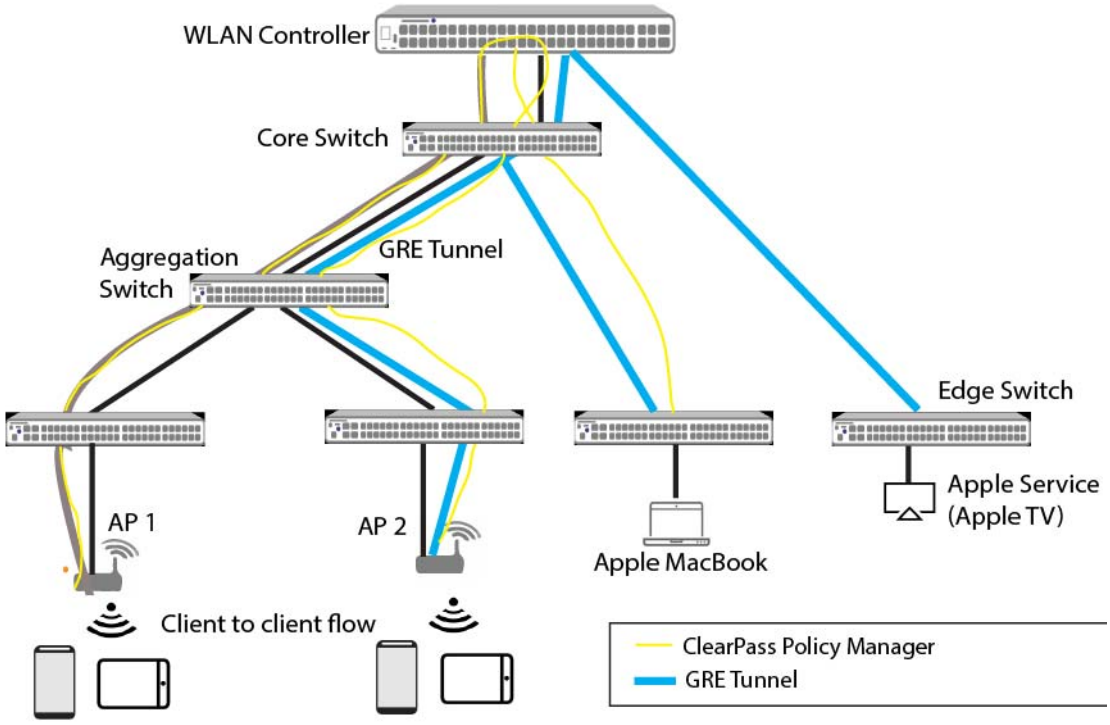


Figure 30-16 : Zero Configuration Networking - Aruba Mode

Gateway Mode

Gateway mode is configured if the network has no WLAN controller. In this mode, the traffic from the edge switch is forwarded to the configured gateway switch.

The gateway will replicate and forward the received mDNS and SSDP packets on all the VLANs, based on a pre-configured VLAN sharing list. Only mDNS and SSDP multicast packets are flooded to the gateway VLAN list. Unicast packets are not flooded to the gateway VLAN list.

The following diagram represents a sample Gateway mode setup:

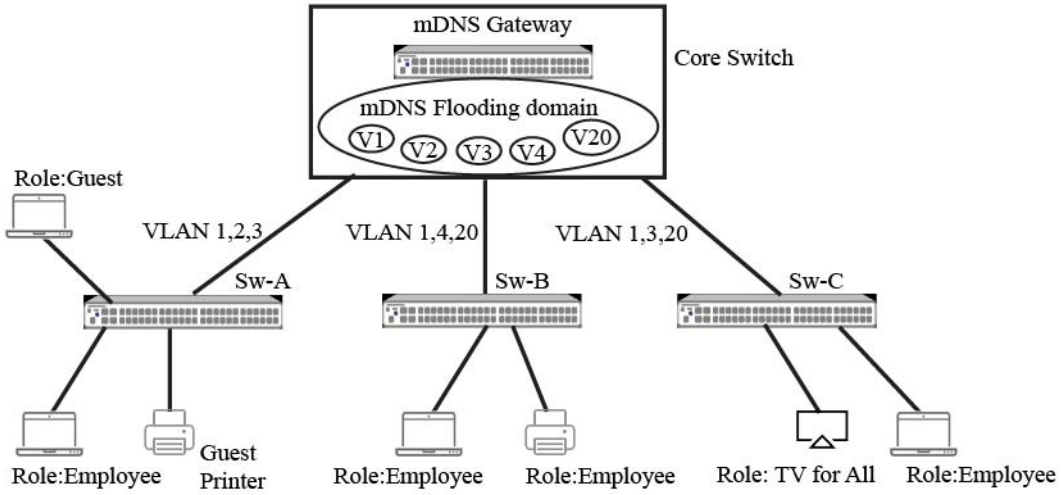


Figure 30-17 : Gateway Mode Setup

The following LAN scenario explains how mDNS service sharing is achieved.

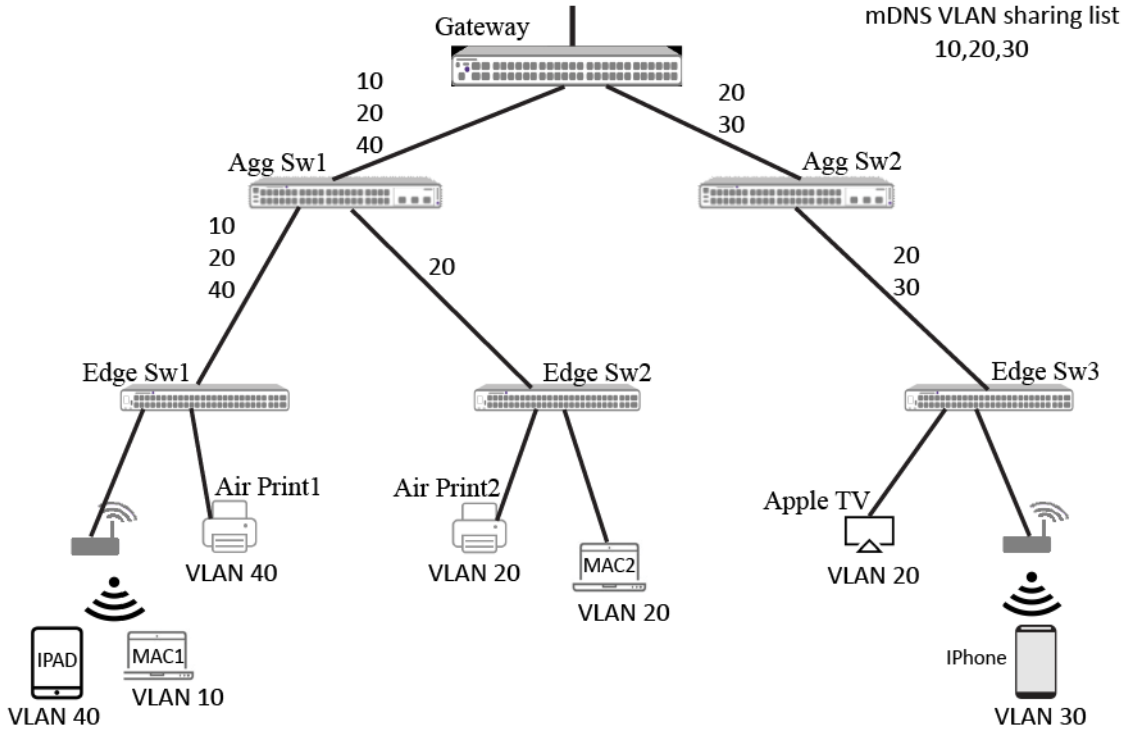


Figure 30-18 : mDNS Service Sharing

Consider a LAN with a mix of wireless and wired users with Apple devices, such as an iPad, iPhone, MAC laptop, Apple TV(Airplay) and Apple printers (AirPrinter) distributed across different VLANs (10,20,30 and 40).

- Airprint1 will advertise its service information through mDNS service advertisement packets.
- Edge switch Sw2 will flood these packets in VLAN20 to the gateway.
- Mac2 laptop in VLAN 20 will receive this advertisement directly from the Edge switch sw2.
- On receiving this mDNS advertisement, the gateway replicates them in each VLAN configured in the VLAN sharing lists (VLAN 10, 20, and 30). So, these advertisements will reach all Apple devices except the guest iPad in VLAN40.
  - Guest iPad will get the service advertisement from Airprint2.
  - All other devices will be able get the print service in Airprint2.
- The mDNS query packet flows are processed in a similar manner.
- Once an Apple device learns the service to IP address mapping, the actual unicast data will be switched or routed depending on the VLANs they are in.

Note. The SSDP service sharing also works similarly.

The following prerequisites apply to this solution:

- All the VLANs that have mDNS clients, SSDP clients, and services must be extended up to the gateway node.
- An IP interface must be configured on the gateway for each of the VLANs.

Gateway Mode: mDNS Relay between SPB and VLAN Domains

In a network where mDNS clients and servers are connected to edge switches that participate in a Shortest Path Bridging (SPB) domain, mDNS traffic is forwarded on an SPB service instance (I-SID) to the gateway switch (mDNS Relay switch). The gateway will replicate and forward the received mDNS packets on all the VLANs, based on a pre-configured VLAN sharing list. This process is similar to how mDNS traffic is forwarded to the gateway switch in a VLAN domain, as shown in "mDNS Service Sharing" on page 30-169.

When mDNS traffic originates from an SPB service domain, an external loopback configuration on the gateway switch provides a method for passing traffic between the SPB and VLAN domains. For example, in the following diagram, mDNS devices are connected to service access ports on SPB Backbone Edge Switches (BEBs). The service access ports are bound to an SPB service to form a Service Access Point (SAP). The mDNS traffic is forwarded on the SAP to the gateway switch where the traffic is passed between the SPB and VLAN domain.

Note. An external loopback on the gateway switch may not be needed if all the mDNS traffic is forwarded on a common SPB service instance. In this case, the mDNS clients and servers would see each other within the SPB domain.

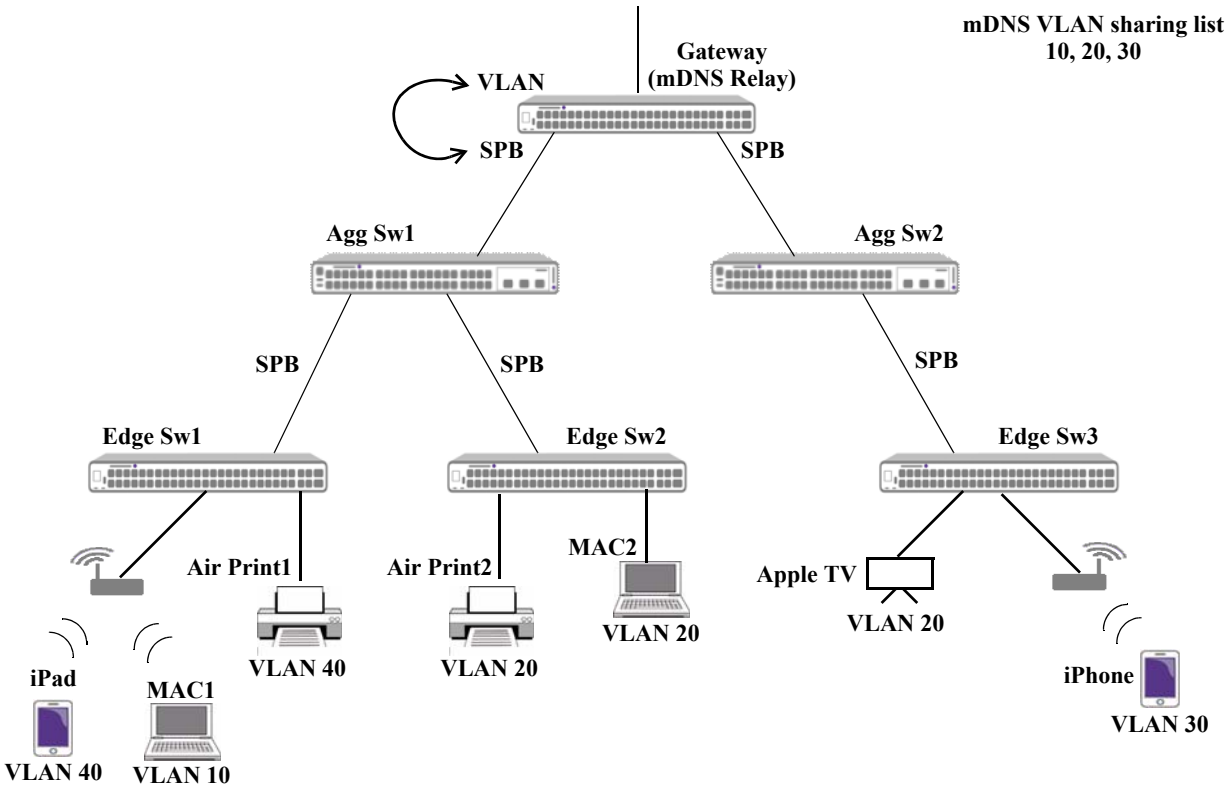


Figure 30-19 : mDNS Gateway - mDNS Devices Connected to an SPB Domain

In the above diagram, all of the mDNS devices are connected to service access ports on SPB BEBs (Edge Sw1, Edge Sw2, and Edge Sw3).

- The service access ports are bound to an SPB service and a VLAN tag to create a virtual SAP port. Traffic received on the access port that has a VLAN tag that matches the tag value of the SAP is encapsulated and forwarded to the gateway through the SPB network backbone.
- When the gateway switch receives encapsulated SPB traffic, the encapsulation is removed and the traffic is passed through the loopback to the VLAN domain.
- The gateway switch will then replicate and flood the received mDNS packets on VLANs 10, 20, 30.

Another supported scenario consists of some mDNS devices connected to an SPB domain and some mDNS devices connected to a VLAN domain. For example, if Edge Sw3 and Agg Sw2 do not participate in the SPB domain, then those switches will forward mDNS traffic through the VLAN domain to the gateway switch. The other SPB switches will forward mDNS traffic through the SPB service domain.

For more information about how to configure an SPB network, see Chapter 7, “Configuring Shortest Path Bridging.”

Standard Mode

Standard mode is configured on the edge switch if the network has an OmniSwtich as the controller. All the edge switches must be mDNS and SSDP enabled. The edge switches must be configured with the L2GRE tunnel with the remote tunnel endpoint IP address of the OmniSwitch controller configured as the responder.

The mDNS and SSDP traffic entering the edge switch is tunneled to the OmniSwitch controller. The mDNS and SSDP traffic received from the controller on the L2GRE tunnel is verified. If the packet is unicast, it is forwarded based on the destination. If the packet is multicast, the packet is flooded to the configured access VLAN list.

Responder Mode

In this mode, Responder is running on an OmniSwitch core switch. The core switch and the edge switches must be mDNS and SSDP enabled. The edge switches must be configured with type standard with the L2GRE tunnel with the remote tunnel endpoint IP address of the OmniSwitch controller configured as the responder.

In this mode, the server policy and client policies are created independently and linked by the service rule. Service sharing rules define the criteria by which the Responder will decide which services can be shared with which client requests. The server and client policy must be configured with at least one of the following attributes: VLAN, Role, Location, Username, or MACaddress. If there are no service rules configured, the Responder learns all the services, but will not process any query which comes from an mDNS or SSDP client.

The following diagram represents a sample Responder mode setup:

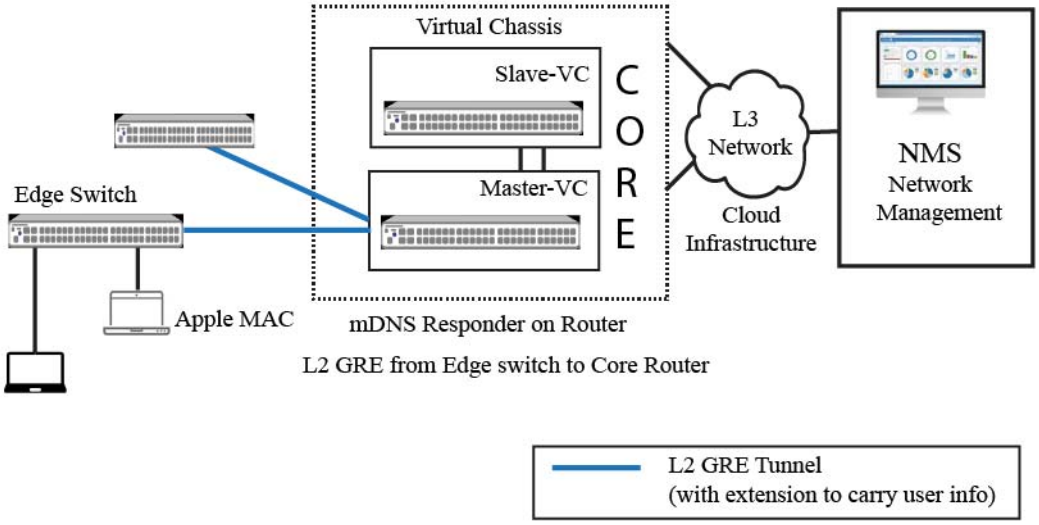


Figure 30-20 : Responder Mode Setup

Responder Mode: mDNS Responder between SPB and VLAN Domains

In a network where mDNS clients and servers are connected to edge switches that participate in a Shortest Path Bridging (SPB) domain, mDNS traffic is forwarded on an SPB service instance (I-SID) to the OmniSwitch mDNS Responder. This process is similar to how mDNS traffic is forwarded to the Responder switch in a VLAN domain, as shown in "Responder Mode Setup" on page 30-172. However, the Responder functionality takes place in the VLAN domain and not in the SPB service domain. As a result, an external loopback configuration on an SPB OmniSwitch that is connected to the Responder switch is required.

The following diagram shows an example in which mDNS devices are connected to service access ports on SPB Backbone Edge Switches (BEBs). The service access ports are bound to an SPB service to form a Service Access Point (SAP). The mDNS traffic is forwarded on the SAP to the Responder switch. The external loopback is configured on the OmniSwitch that is connected to the Responder switch. This is where the mDNS traffic is passed between the SPB and VLAN domain and then tunneled to the OmniSwitch Responder.

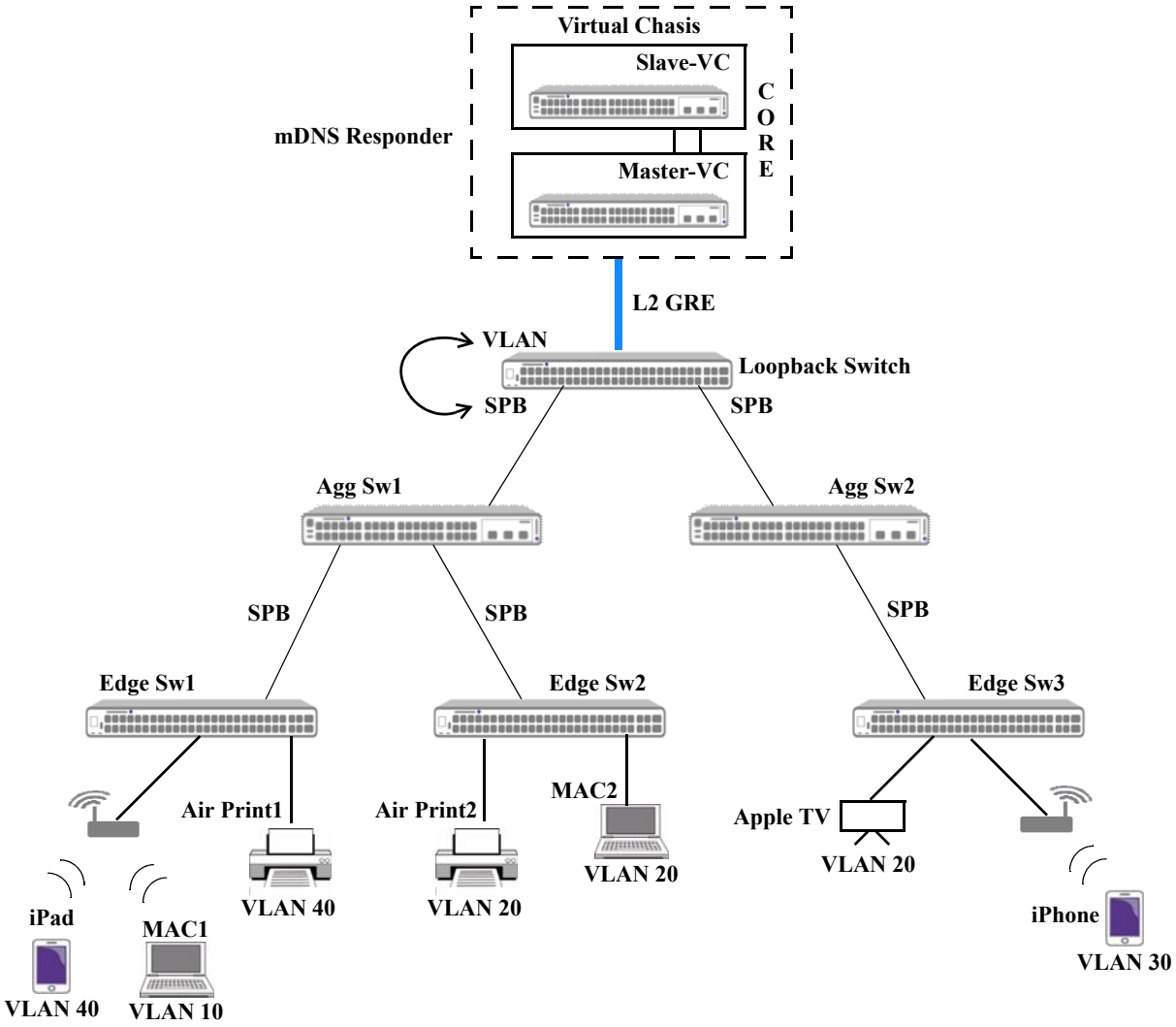


Figure 30-21 : mDNS Responder - mDNS Devices Connected to an SPB Domain

In the above diagram, all of the mDNS devices are connected to service access ports on SPB BEBs (Edge Sw1, Edge Sw2, and Edge Sw3).

- The service access ports are bound to an SPB service and a VLAN tag to create a virtual SAP port. Traffic received on the access port that has a VLAN tag that matches the tag value of the SAP is encapsulated and forwarded through the SPB network backbone.
- When the Loopback Switch receives encapsulated SPB traffic, the encapsulation is removed and the traffic is passed through the loopback to the VLAN domain.
- The mDNS packets are then forwarded through an L2 GRE tunnel that connects the Loopback Switch to the mDNS Responder switch, where service sharing rules are applied.

Another supported scenario consists of some mDNS devices connected to an SPB domain and some mDNS devices connected to a VLAN domain. For example, if Edge Sw3 and Agg Sw2 do not participate in the SPB domain, then those switches will forward mDNS traffic through an L2 GRE tunnel in the VLAN domain to the mDNS Responder switch. The other SPB switches will forward mDNS traffic through the SPB service domain.

For more information about how to configure an SPB network, see Chapter 7, "Configuring Shortest Path Bridging."

Note. Refer to "Quick Steps for Zero Configuration mDNS and SSDP" on page 30-174 for procedures on how to configure each mode.

Backward Compatibility

mDNS and SSDP with Aruba mode is backward compatible. The configurations made using the older CLI commands will continue to work provided the Loopback0 IP address is configured as the source endpoint address. Responder IP will be set to destination IP of the legacy tunnel IP interface configuration. Any further changes to the existing configuration can be made only after saving the configuration in the new format using the **write memory** command.

Configuration Guidelines

- The mDNS or SSDP packet handling across Layer 3 supports different modes of operation, such as Tunnel (relay to Aruba), Gateway, Tunnel Standard (relay to OmniSwitch responder), and Responder mode. At any given point of time, the switch can operate in only one mode.
- The operational status of mDNS and SSDP on the edge switch will be down if the responder IP address is not configured. The responder IP address configured must be reachable.
- The Loopback0 IP address must be configured on the edge switch and reachable from the controller.
- When the OmniSwitch is running as a responder (Responder mode), the query packets are flooded through only those VLANs that are configured in the access VLAN list on the edge switch.
- When mDNS relay or SSDP relay status is disabled, the mDNS or SSDP packets are dropped.
- The responder triggers a service request query in the event of takeover, VC takeover, reload, and responder initiated query requests. Configuring the known service list is recommended. The responder queries only those services that are configured in the service list.

Quick Steps for Zero Configuration mDNS and SSDP

Zero configuration varies with respect to the type of modes (Tunnel (Aruba), Gateway, Tunnel Standard, Responder Mode).

If the network consists of Aruba wireless controllers (Aruba mode), the following must be configured on the edge switch:

1 Enable mDNS and SSDP functionality using the **zeroconf mdns admin-state** and **zeroconf ssdp admin-state** command.

To enable mDNS relay on the switch, enter:

```
-> zeroconf mdns admin-state enable
```

To enable SSDP relay on the switch, enter:

```
-> zeroconf ssdp admin-state enable
```

2 Configure the mode of operation for the switch. For Aruba APs, the mode must be set to tunnel. To configure the mode, use the **zeroconf mode** command. For example:

```
-> zeroconf mode tunnel
```

Note. By default, the switch will be in tunnel mode (Aruba mode).

3 Configure the tunnel source IP address (Loopback0 IP interface) for the GRE tunnel. To configure the Loopback0 IP interface, use the **ip interface** command. For example:

```
-> ip interface Loopback0 address 10.1.2.3
```

Note. Ensure reachability to the Loopback0 IP address. If the Loopback0 address is not configured, the operational status of mDNS or SSDP will be down.

4 Configure the Aruba responder IP address. The responder IP address must be configured to tunnel the mDNS or SSDP packets in the tunnel mode. To configure the responder IP address, use the **zeroconf responder-ip** command. For example:

```
-> zeroconf responder-ip 10.0.1.5
```

Note. If the responder IP address is not configured or not reachable, then the operational status of mDNS or SSDP will be down.

If the network does not consist of any responder and uses a gateway, the following must be configured on the edge switch:

1 Enable mDNS and SSDP functionality using the **zeroconf mdns admin-state** and **zeroconf ssdp admin-state** command.

To enable mDNS relay on the switch, enter:

```
-> zeroconf mdns admin-state enable
```

To enable SSDP relay on the switch, enter:

```
-> zeroconf ssdp admin-state enable
```

2 Configure the mode of operation for the switch. For gateway mode, set the mode to gateway. To configure the mode, use the **zeroconf mode** command. For example:

```
-> zeroconf mode gateway
```

3 Configure the gateway VLAN list. Traffic from the edge switches will be forwarded at L2 to the gateway switch. From the gateway switch, the mDNS and SSDP packets will be relayed to other VLANs based on the gateway VLAN list configured. To configure the gateway VLAN list, use the **zeroconf gateway-vlan-list** command. For example:

```
-> zeroconf gateway-vlan-list 1 4 6
```

Note. Maximum of 10 gateway VLANs is supported in a list.

If the network consists of an OmniSwitch controller as responder (Tunnel Standard mode), the following must be configured on the edge switch:

1 Enable mDNS and SSDP functionality using the **zeroconf mdns admin-state** and **zeroconf ssdp admin-state** command.

To enable mDNS relay on the switch, enter:

```
-> zeroconf mdns admin-state enable
```

To enable SSDP relay on the switch, enter:

```
-> zeroconf ssdp admin-state enable
```

2 Configure the mode of operation for the switch. Set the mode to tunnel type standard. To configure the mode, use the **zeroconf mode** command. For example:

```
-> zeroconf mode tunnel type standard
```

3 Configure the tunnel source IP address (Loopback0 IP interface) for the GRE tunnel. To configure the Loopback 0 IP interface, use the **ip interface** command. For example:

```
-> ip interface Loopback0 address 10.1.2.3
```

Note. Ensure reachability to the Loopback0 IP address. If the Loopback0 address is not configured, the operational status of mDNS or SSDP will be down.

4 Configure the remote tunnel endpoint IP address of the switch running as the responder. To configure the responder IP address, use the **zeroconf responder-ip** command. For example:

```
-> zeroconf responder-ip 10.0.1.5
```

5 Configure the access VLAN list. For any query packet generated by the responder, mDNS packet comes in with a VLAN tag of 4095. These packets are flooded to the configured access VLAN list. To configure the access VLAN list, use the **zeroconf access-vlan-list** command. For example:

```
-> zeroconf access-vlan-list 7 8 9
```

Note. Maximum of 16 access VLANs is supported in a list.

When Responder is running on a core OmniSwitch (Responder mode), the following must be configured on the responder:

1 Enable mDNS and SSDP functionality using the **zeroconf mdns admin-state** and **zeroconf ssdp admin-state** command.

To enable mDNS relay on the switch, enter:

```
-> zeroconf mdns admin-state enable
```

To enable SSDP relay on the switch, enter:

```
-> zeroconf ssdp admin-state enable
```

2 Configure the mode of operation for the switch. For responder mode, set the mode to responder. To configure the mode, use the **zeroconf mode** command. For example:

```
-> zeroconf mode responder
```

3 In responder mode, the tunnel for each edge switch can be configured manually or set to auto mode configuration. Use the **zeroconf edge-ip-list mode** command to configure the edge IP mode as manual or auto. For example:

```
-> zeroconf edge-ip-list mode auto
```

In **auto** mode, the edge-IPs are automatically learned by the responder.

```
-> zeroconf edge-ip-list mode manual
```

In **manual** mode, the edge-IPs has to be configured manually.

4 To configure the tunnel for the edge switch, use the **zeroconf edge-ip-list** command. For example:

```
-> zeroconf edge-ip-list 10.1.2.3 10.1.2.4
```

5 Configure the service rules and link the server policies and client policies to the service rule. Service rules define the criteria by which the mDNS responder will decide the services that can be shared with the clients.

- To configure the server policy, use the **zeroconf server-policy** command. Provide the server role, VLAN ID, location name, username or MAC address to be mapped with server policy using this command. A server policy must be configured either with a role, VLAN ID, location, username or MAC address attribute. At least one attribute must be configured. For example:

```
-> zeroconf server-policy SP1 role employee
```

- To configure the client policy, use the **zeroconf client-policy** command. Provide the client role, VLAN ID, location name, username or MAC address to be mapped with client policy using this command. A client policy must be configured either with a role, VLAN ID, location, username or MAC address attribute. At least one attribute must be configured. For example:

```
-> zeroconf client-policy CP1 vlan 10 20 30
```

- To map the configured server policy and client policy with the service rule, use **zeroconf service-rule policy** command. There must be one-to-one mapping between the server and client policy. For example:

```
-> zeroconf service-rule SR1 server-policy SP1 client-policy CP1
```

6 Service rules must be mapped to the supported service IDs. To map the selective services for a service rule to be shared with the clients, use **zeroconf service-rule service-id** command. For example,

To map the mDNS services to the service rule, enter:

```
-> zeroconf service-rule SR1 mdns-service-id _ipp._tcp.local _scanner._tcp.local
```

To map the SSDP services to the service rule, enter:

```
-> zeroconf service-rule SR1 ssdp-service-id urn:schemas-upnp-
org:device:MediaServer:1 upnp:rootdevice
```

Service Rule Configuration

Service rules are configured on the responder switch. Service rules define the criteria by which the mDNS responder will decide the services that can be shared with the clients. If there are no service rules configured, the responder switch will learn all the services but will not process any query received from the mDNS or SSDP client.

Configure the service rules and link the server policies and client policies to the service rule. Each server and client policy must contain at least a role, VLAN ID, location, username or MAC address. A server or client policy can contain up to 16 roles, 16 locations, 16 VLANs, 16 usernames, and 16 MAC addresses. There must be one-to-one mapping between server and client policy. To map the configured server policy and client policy with the service rule, use the **zeroconf service-rule policy** command. For example:

```
-> zeroconf service-rule SR1 server-policy SP1 client-policy CP1
```

Service rules must be mapped to the selective services to be shared with the clients. The selective service is configured by mapping the Service IDs to the service rule. A service ID could be part of multiple server rules. To map the service ID to a service rule, use the **zeroconf service-rule service-id** command. For example,

To map the mDNS services to the service rule, enter:

```
-> zeroconf service-rule SR1 mdns-service-id _ipp._tcp.local _scanner._tcp.local
```

To map the SSDP services to the service rule, enter:

```
-> zeroconf service-rule SR1 ssdp-service-id urn:schemas-upnp-
org:device:MediaServer:1 upnp:rootdevice
```

Service List Configuration

A service list contains a list of configured service IDs. A service ID can be a configured service ID or a learned service ID. Service IDs that belong to a service list are called configured service IDs. The learned services are the services which are received by the responder but not part of the configured service list.

The service list can contain a maximum of 64 service IDs.

The mDNS or SSDP responder will query and re-learn the configured services in the service list in the event a takeover, VC takeover, reload and responder initiated query request occurs.

To configure a list of known services for mDNS service query, use the **zeroconf service-list** command. For example:

```
-> zeroconf mdns service-list _ipp._tcp.local _scanner._tcp.local
```

To configure a list of known services for SSDP service query, use the **zeroconf service-list** command. For example:

```
-> zeroconf ssdp service-list urn:schemas-upnp-org:device:MediaServer:1
upnp:rootdevice
```

Note. The learned services will not be re-learned by the responder. The service must be added in the service list for re-learning.

Learning a Service

If the responder doesn't learn a service, that specific service can be queried for re-learning by the responder. To manually query for a specific service, use the **zeroconf service-id query-request** command. For example,

Learning an mDNS service:

```
-> zeroconf mdns service-id _scanner._tcp.local query-request
```

Learning an SSDP service:

```
-> zeroconf ssdp service-id upnp:rootdevice query-request
```

Refreshing the mDNS or SSDP Database

The mDNS or SSDP database can be manually refreshed to update the latest server policies, client policies, server rules, learned service instances and service cache entry. Use the **zeroconf refresh-database** command to refresh the mDNS or SSDP database. For example,

To refresh the mDNS database:

```
-> zeroconf mdns refresh-database
```

To refresh the SSDP database:

```
-> zeroconf ssdp refresh-database
```

After refreshing the database, use the **show zeroconf services-cache** and **show zeroconf server policy-instances** commands to display the latest cache and service instances information.

Verifying the Zero Configuration

To verify the Zero configuration on the switch, use the following show command:

show zeroconf	Displays the basic zero configuration details.
show zeroconf services	Displays all the configured mDNS or SSDP service IDs.
show zeroconf services-cache	Display the list of services learned by mDNS or SSDP stack.
show zeroconf edge-details	Displays the IP address and reachability status of the tunnel endpoints configured on the responder.
show zeroconf server policies	Displays the information of the server policies configured.
show zeroconf client policies	Displays the information of the client policies configured on the responder.
show zeroconf service rules	Displays the information of the service rules configured on the responder.
show zeroconf server policy-instances	Displays the mDNS or SSDP service instance learned on the responder for each server policy.

Note. For more information on the CLI command usage, refer to the *OmniSwitch AOS Release 8 CLI Reference Guide*.

BYOD Application Examples

Note. The application examples provided in this section are specific to the ClearPass Policy Manager (CPPM). Refer to the OmniVista Unified Policy Access Manager (UPAM) documentation for in-depth configuration information and requirements.

The application scenarios provide various examples of how the ClearPass server and the OmniSwitch can be leveraged to provide different network access levels and Universal Network Profiles (UNPs) for employees, guests, and other network-based devices.

In the following contexts the main parameters (such as a UNP name, VLAN ID, and other parameters specified in the application examples), are as follows:

Employee Registered Device — 802.1X Authentication

- UNP = UNP-employee
- VLAN = 96

IP Phone — MAC Authentication

- UNP = UNP-phone
- VLAN = 1002

Guest Device —MAC Authentication with Guest Login

- Registration UNP = UNP-restricted
- Registration VLAN = 96
- Redirect Server = 10.255.95.206
- Guest UNP = UNP-guest
- Guest VLAN = 96

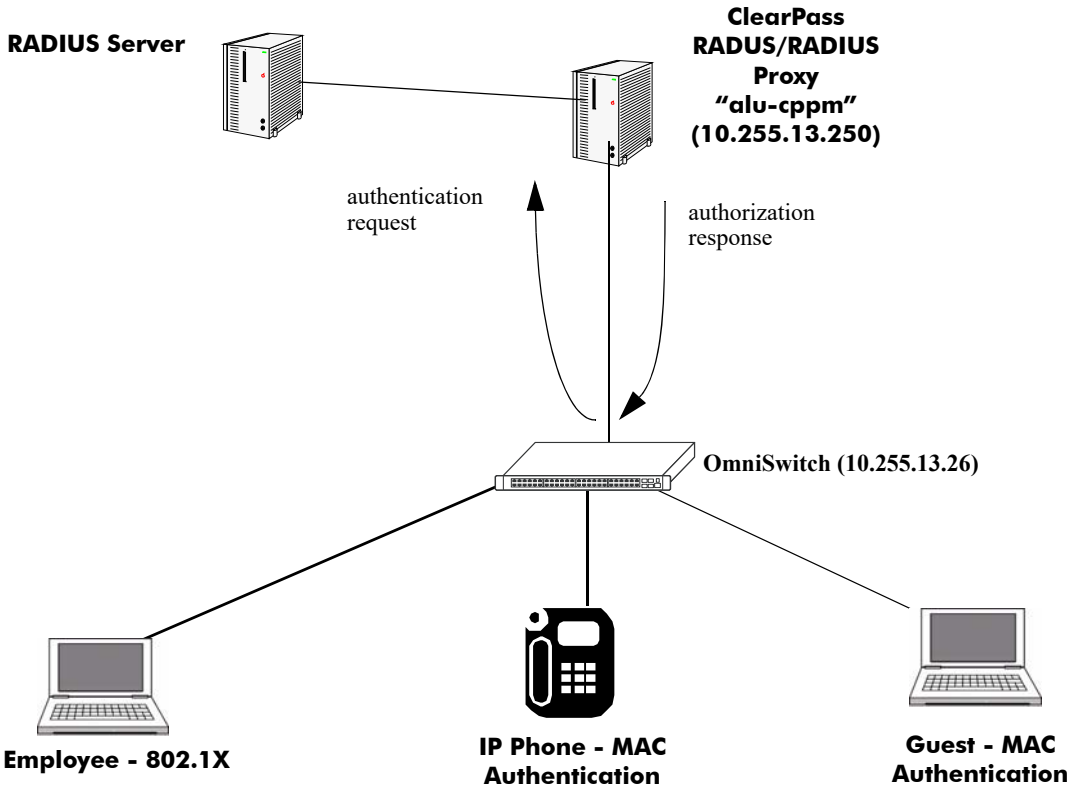


Figure 30-22 : BYOD Network with Employee and Guest Devices

Application Example 1: 802.1X — OmniSwitch Configuration

The OmniSwitch configuration for an 802.1X supplicant:

1 Enable UNP port-based functionality as follows:

```
-> unp port 1/1/11 port-type bridge
-> unp port 1/1/11 802.1x-authentication
```

2 Configure 802.1X authentication for ClearPass RADIUS on the OmniSwitch as follows:

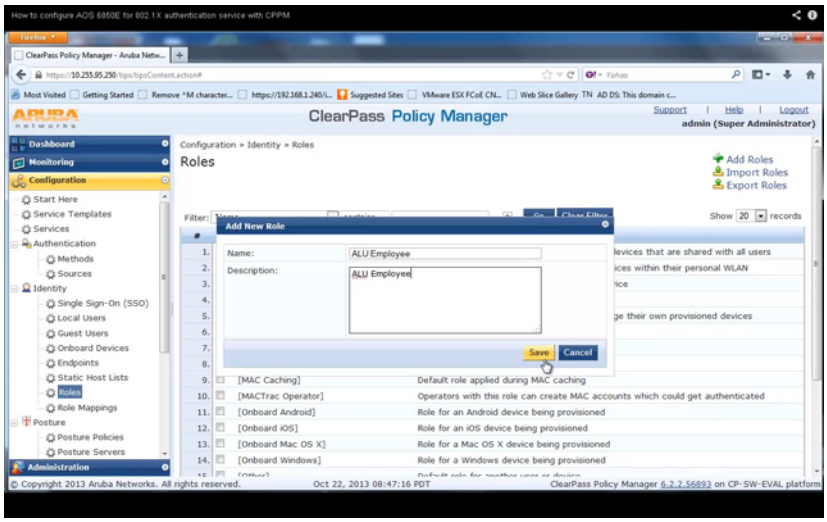
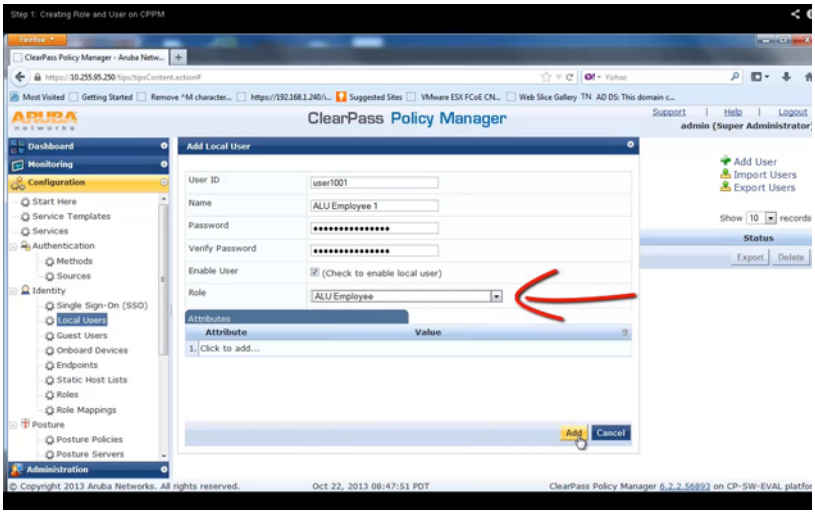
```
-> aaa radius-server alu-cppm host 10.255.95.250 key ale
-> aaa device-authentication 802.1x alu-cppm
```

3 Configure the UNP profile as follows:

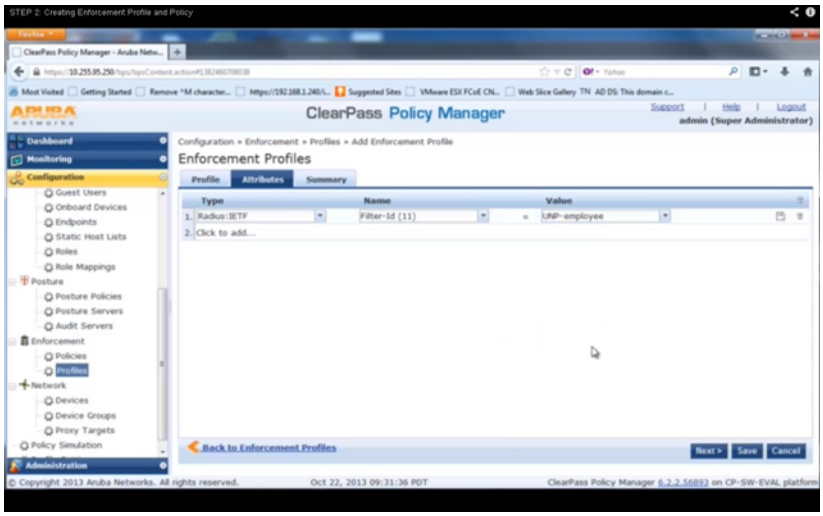
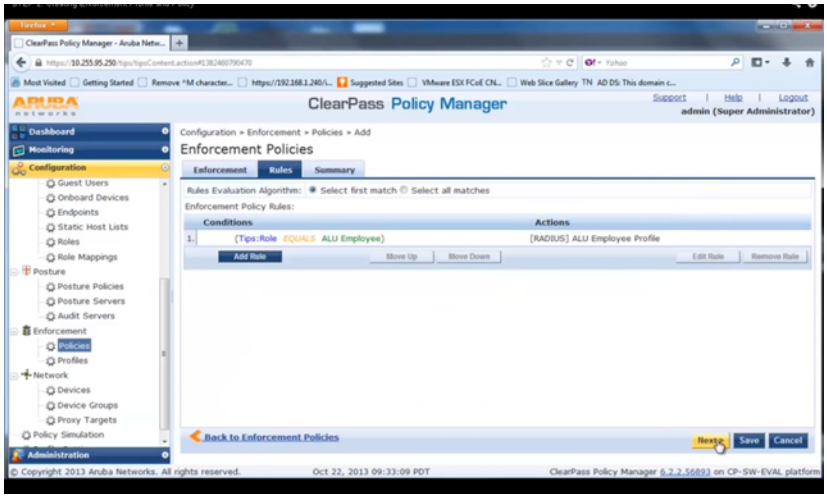
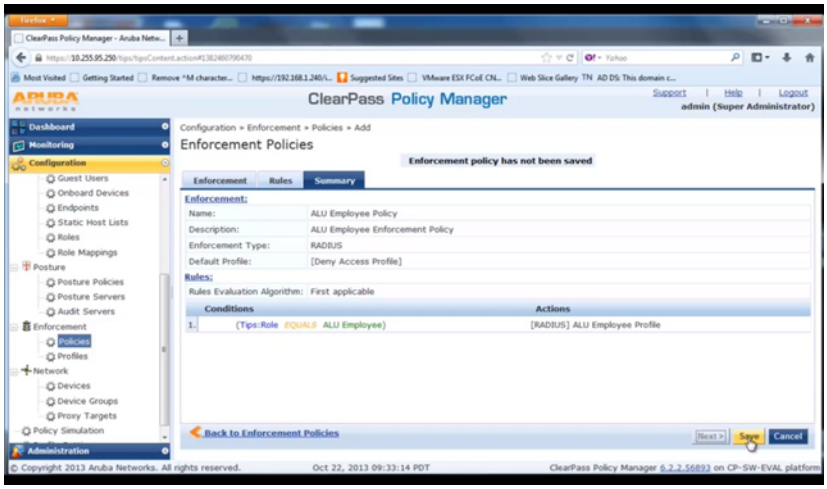
```
-> unp profile UNP-employee
-> unp profile UNP-employee map vlan 96
```

Application Example 1: 802.1X — ClearPass Configuration

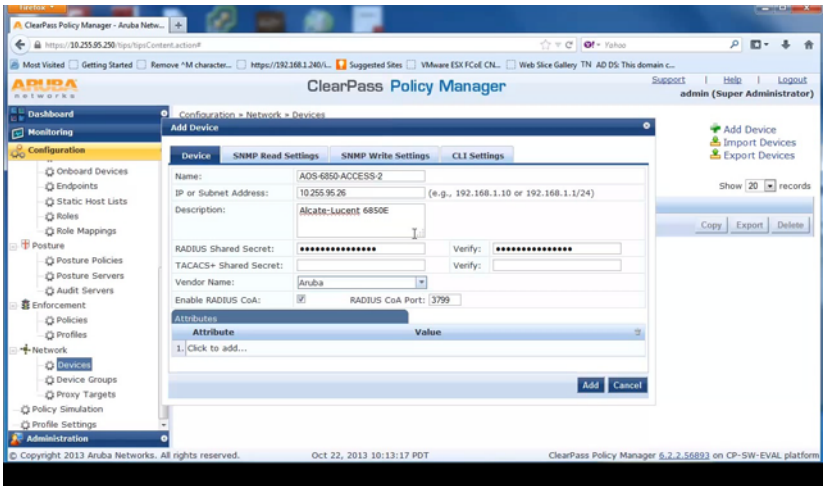
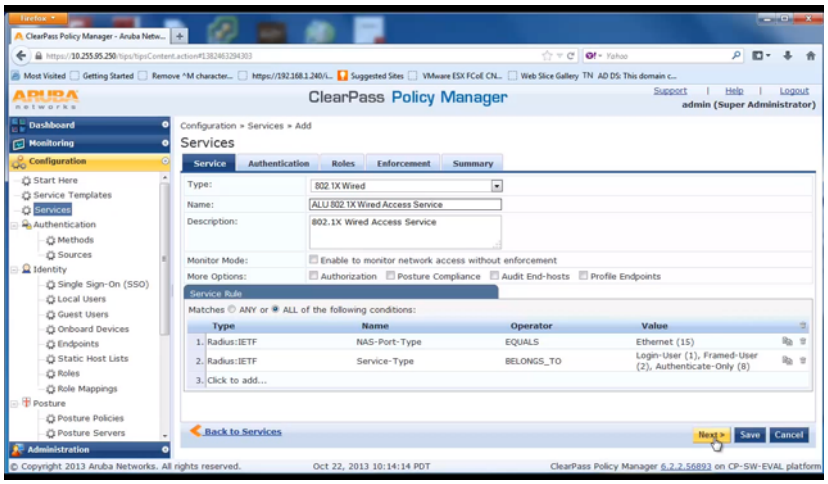
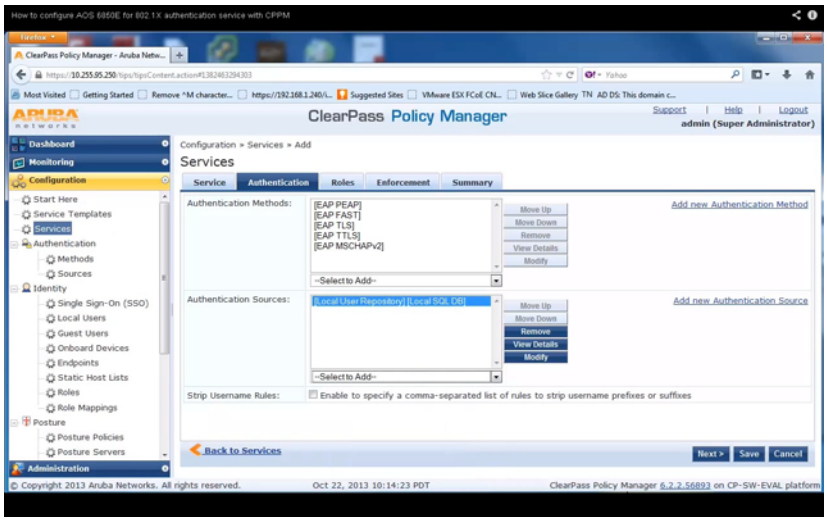
Step 1. ClearPass (802.1X) - Creating employee users and roles

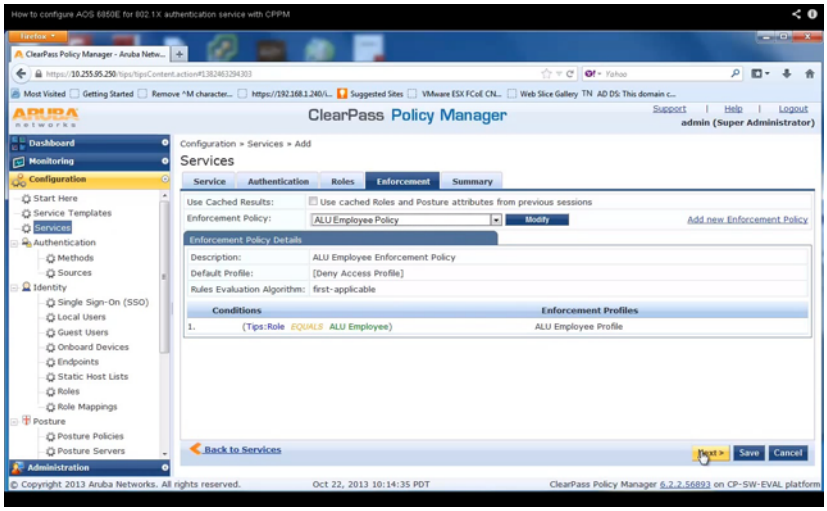
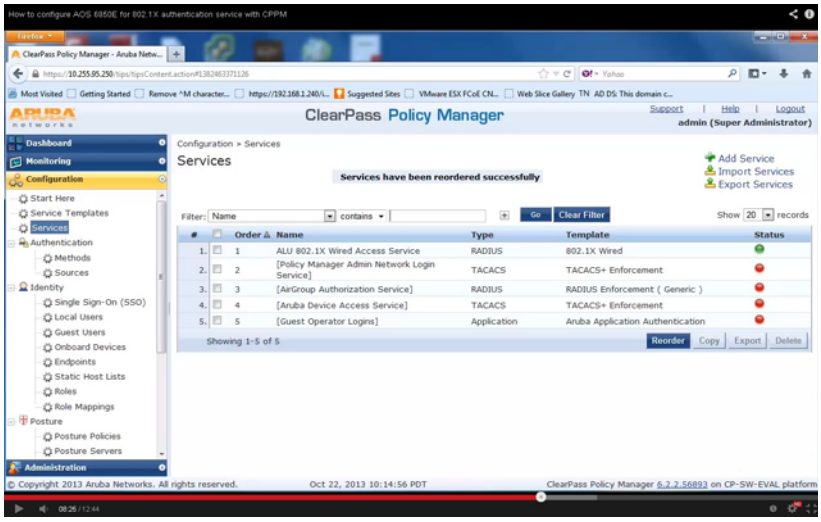
Create user role: **Roles->Add Roles**	
Create users and assign role: **Local Users -> Add Users**	

Step 2. ClearPass (802.1X) - Create Profiles and Policies

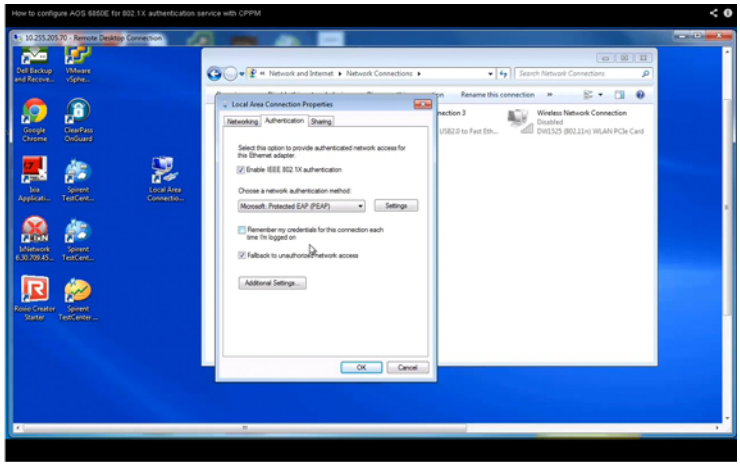
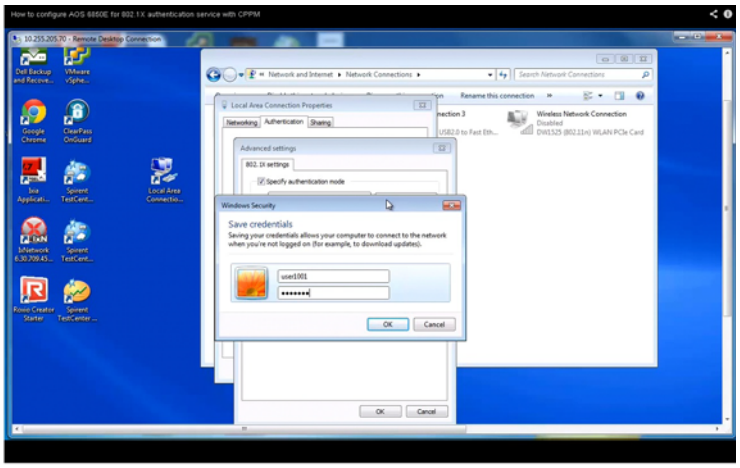
Create Profile: **Attributes** (tab) - **Type:** Radius:IETF - **Filter-ID** (11) - **Value** = UNP-employee (Note: must match UNP Profile on OmniSwitch)	
Create Enforcement Policy: **Rules (tab)**	
View Policies Summary **Summary**	

Step 3. ClearPass (802.1X) - Create 802.1X services

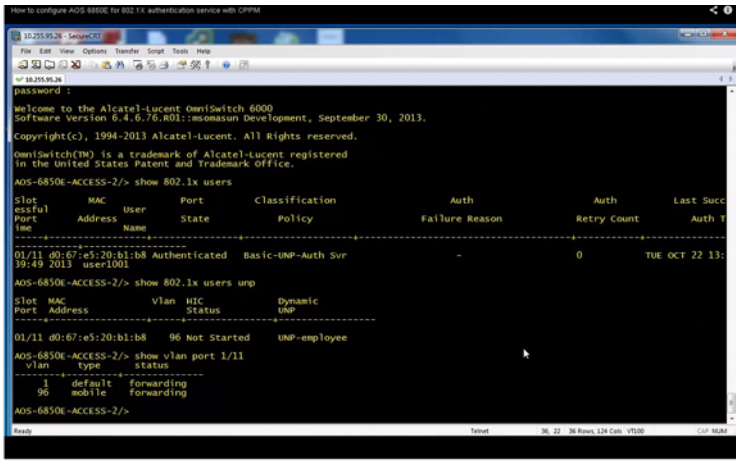
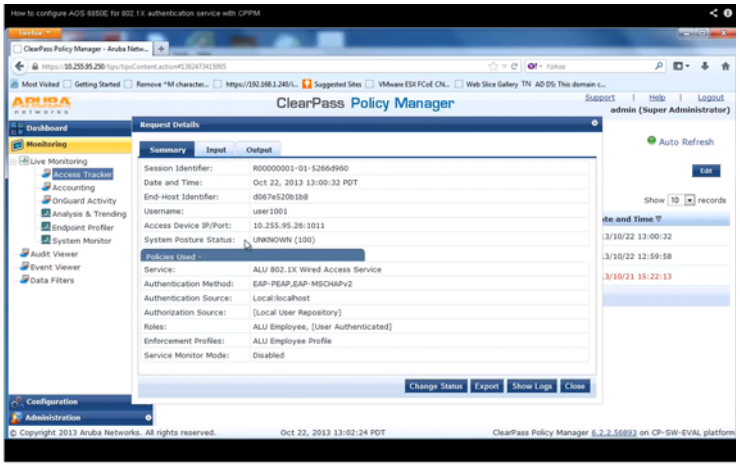
Add OmniSwitch to ClearPass Database **Devices** (tab)	
Add 802.1X Wired Service **Service** (tab)	
Configure 802.1X Authentication **Authentication** (tab)	

Configure Enforcement **Enforcement** (tab)	
Reorder Authentication Devices **Devices**	

Step 4. ClearPass (802.1X) - Configure PC

Configure PC **Properties**	
Configure PC **Advanced Settings**	

Step 5. ClearPass (802.1X) - Confirm Device Authentication

Confirm device Authentication **OmniSwitch**	
Confirm Device Authentication **ClearPass**	

Application Example 2: IP Phone — OmniSwitch Configuration

The OmniSwitch configuration for a non-supplicant IP phone:

1 Configure UNP port-based functionality as follows:

```
-> unp port 1/1/13 port-type bridge
-> unp port 1/1/13 mac-authentication
```

2 Configure MAC authentication for ClearPass RADIUS on an OmniSwitch as follows:

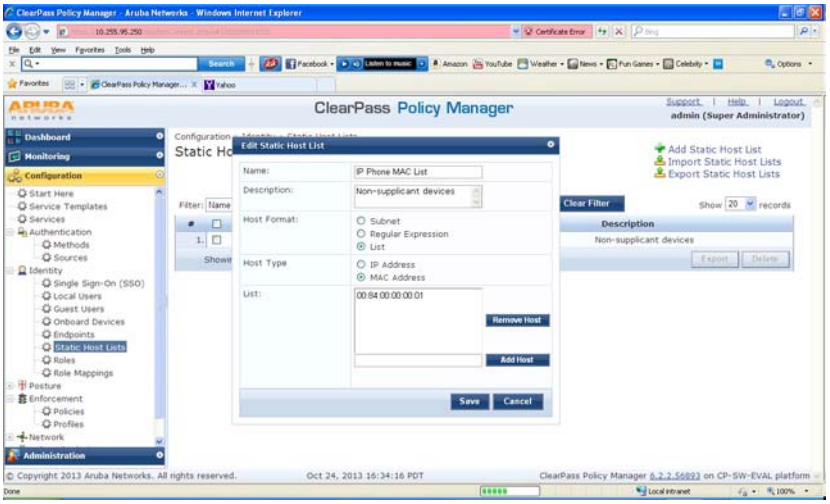
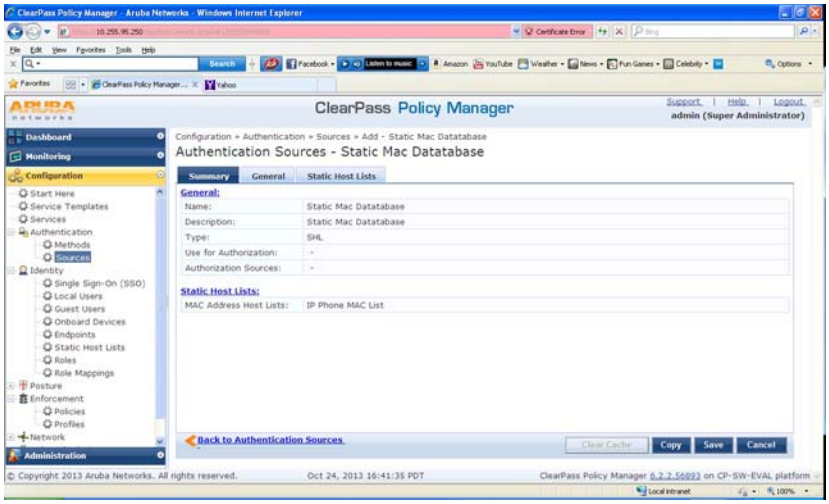
```
-> aaa radius-server alu-cppm host 10.255.95.250 key ale
-> aaa device-authentication mac alu-cppm
```

3 Configure the UNP profile as follows:

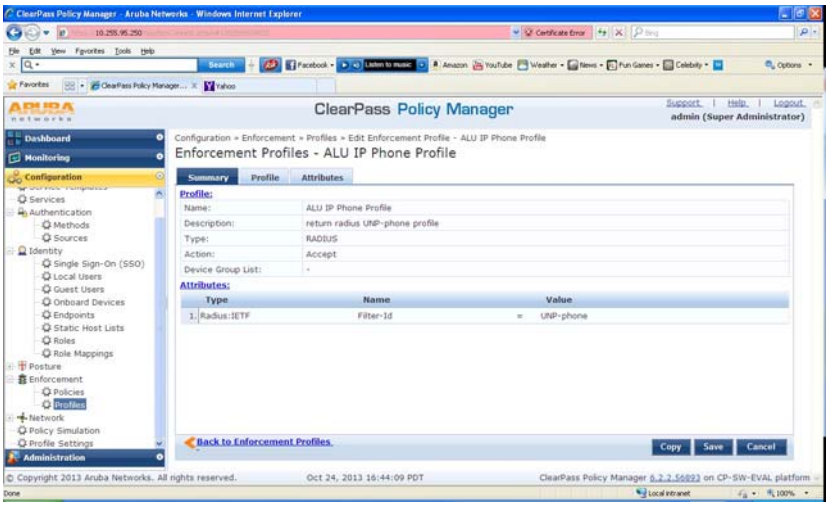
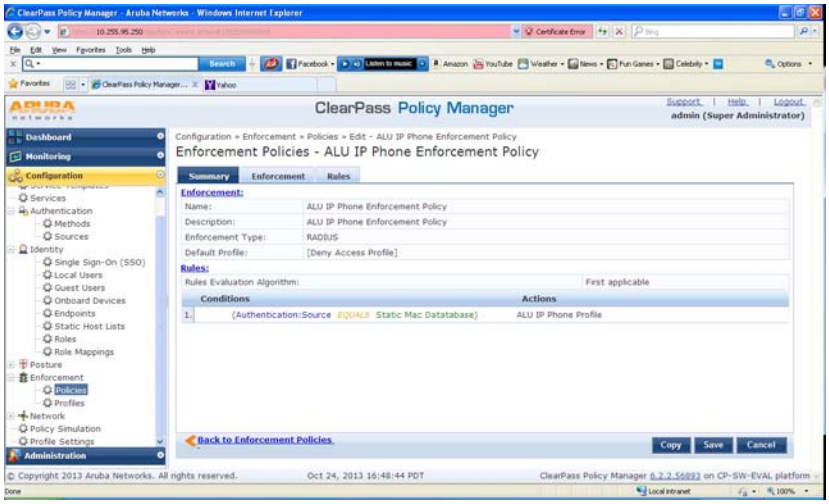
```
-> unp profile UNP-phone
-> unp profile UNP-phone map vlan 1002
```

Application Example 2: IP Phone — ClearPass Configuration

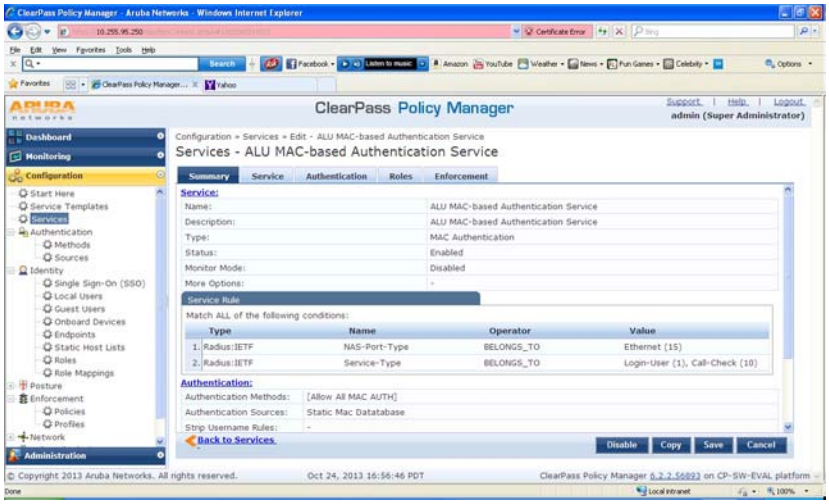
Step 1. ClearPass (IP Phone) - Creating static host list

Create static host list: **Identity->Static Host List**	
Create Authentication Source **Authentication-Sources-Add Authentication Source** **Type**: Static Host List **Host List**: IP Phone MAC List	

Step 2. ClearPass (IP Phone) - Create Profiles and Policies

Create Profile: **Profile (tab)** - **Name:** ALU IP Phone Profile - **Template:** Aruba RADIUS Enforcement **Attributes** (tab) - **Type:** Radius:IETF - **Filter-ID** (11) - **Value** = UNP-phone (Note: must match UNP Profile on OmniSwitch)	
Create Enforcement Policy: **Rules (tab)** - **Type:** Authentication - **Name:** Source - **Operator:** EQUALS - **Value:** Static Mac Database - **Profile Name:** ALU IP Phone Profile	

Step 3. ClearPass (IP Phone) - Create MAC Authentication Service

Add MAC Authentication Service **Service** (tab) **-Type:** MAC Authentication **Authentication** (tab) **- Authentication Sources:** Static MAC Database **Enforcement** (tab) **- Enforcement Policy:** ALU IP Phone Enforcement Policy	

Application Example 3: Guest — OmniSwitch Configuration

The OmniSwitch configuration for guest UNP, VLANs, and redirection:

1. Configure UNP port-based functionality as follows:

```
-> unp port 1/1/13 port-type bridge
-> unp port 1/1/13 802.1x-authentication
-> unp port 1/1/13 mac-authentication
```

2. Configure MAC-authentication for ClearPass RADIUS on an OmniSwitch as follows:

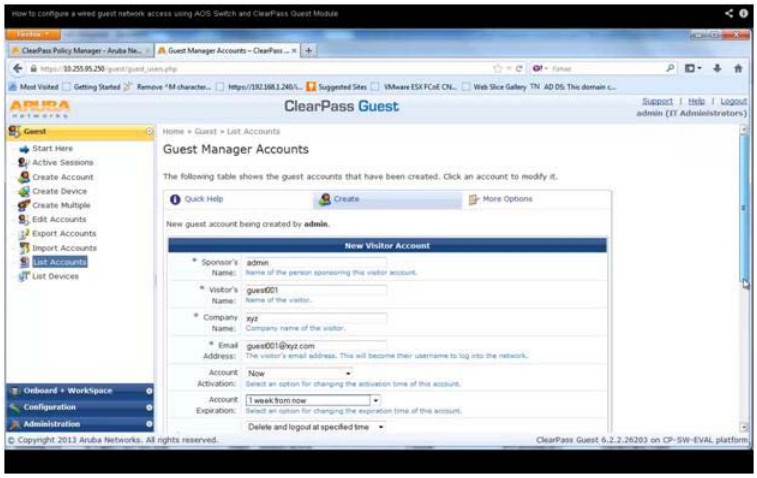
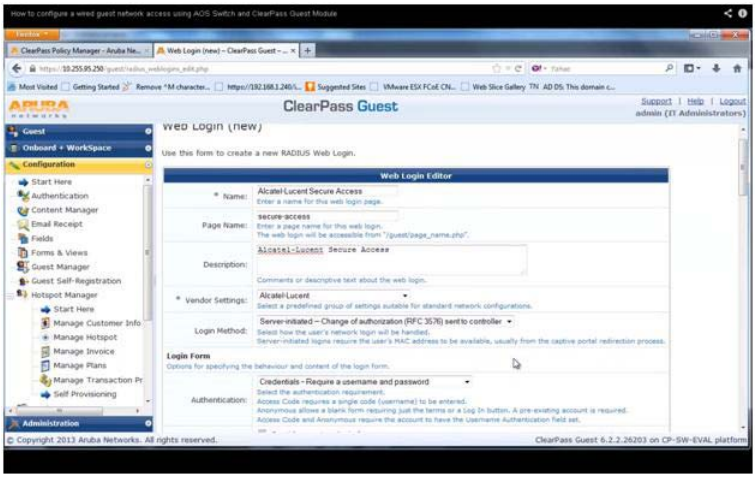
```
-> aaa radius-server alu-cppm host 10.255.95.250 key ale
-> aaa device-authentication 802.1x alu-cppm
-> aaa device-authentication mac alu-cppm
-> aaa accounting 802.1x alu-cppm
-> aaa accounting mac alu-cppm
```

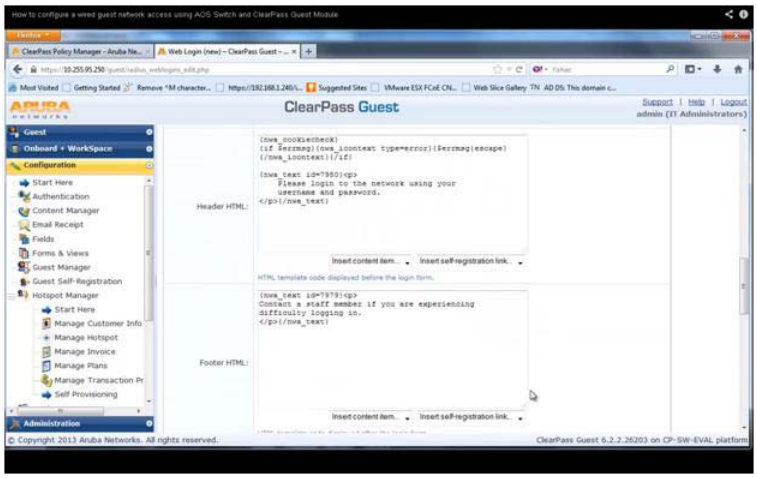
3. Configure UNPs and redirect server as follows:

```
-> unp profile UNP-guest
-> unp profile UNP-guest map vlan 96

-> unp profile UNP-restricted
-> unp profile UNP-restricted map vlan 96

-> unp redirect server ip-address 10.255.95.250
```

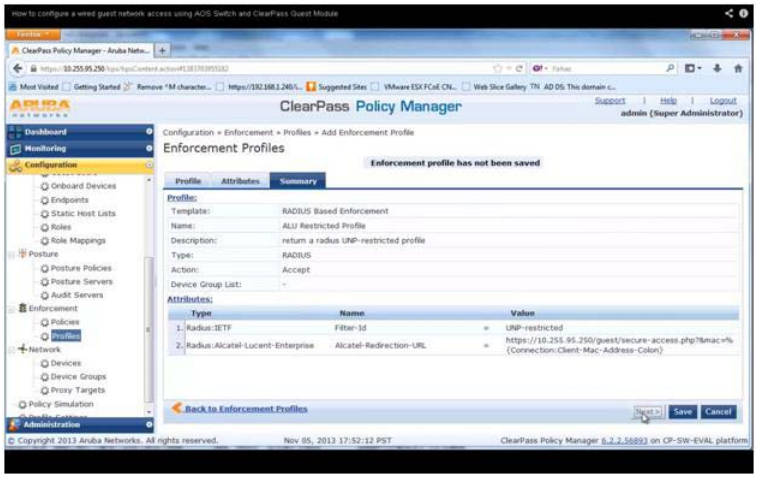
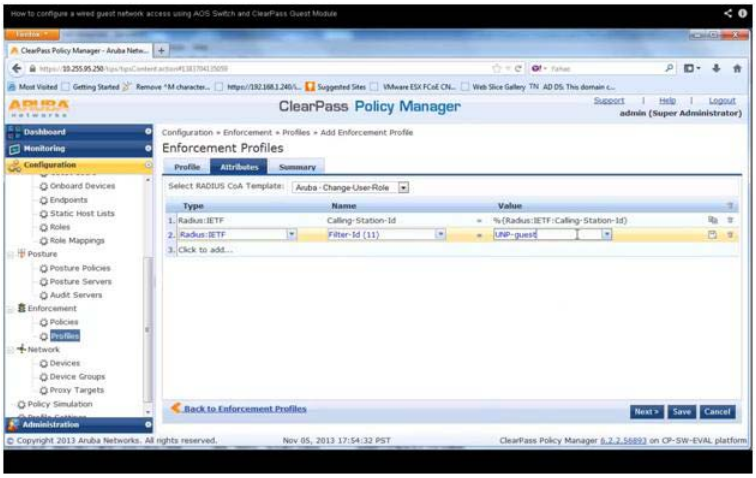
Application Example 3: Guest — ClearPass Configuration

Step 1: ClearPass (Guest) - Create Guest Account and Web login page

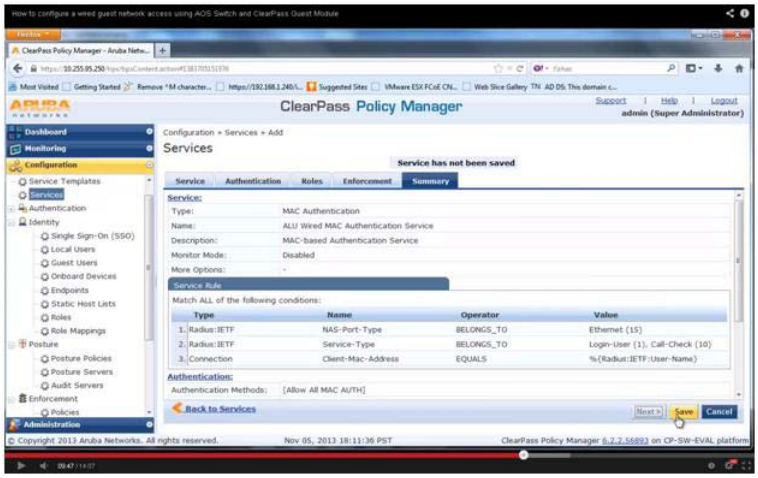
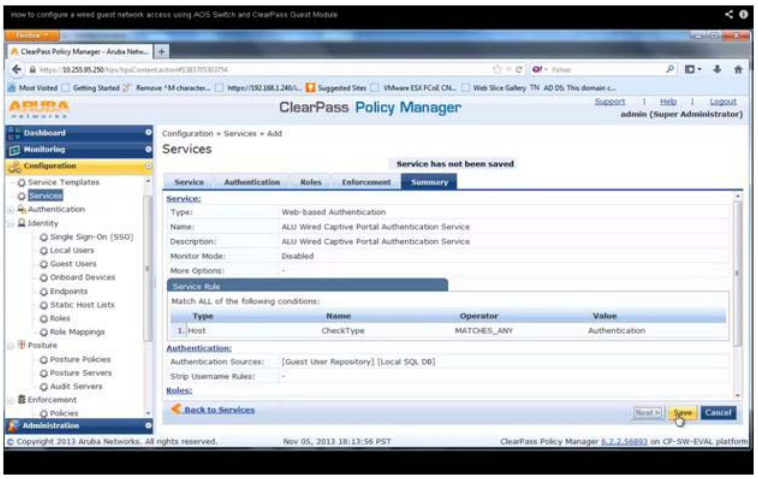
Create guest account **Guest->List Accounts**	
Create web login page **Configuration-Web Logins** **Name**- Alcatel-Lucent Secure Access **Page name:** secure-access **Vendor Settings:** Alcatel-Lucent **Login Method:** Server-initiated **Pre-Auth Check:**None **Terms:** checked **Default URL:** www.google.com **Override Destination:** checked	

Create custom skin if desired	

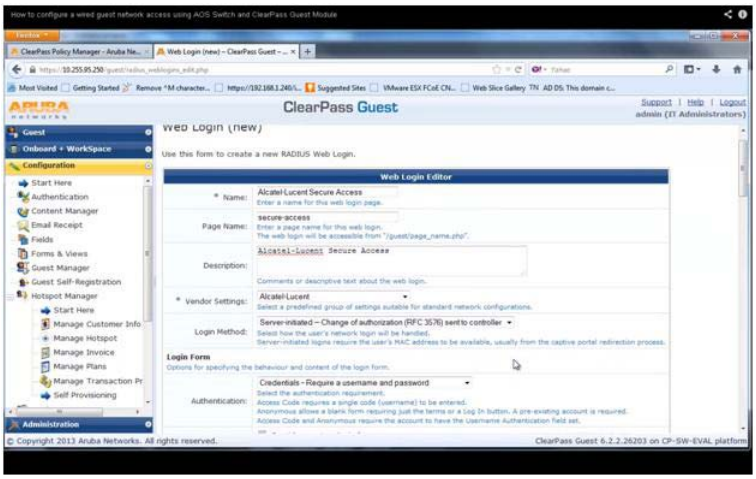
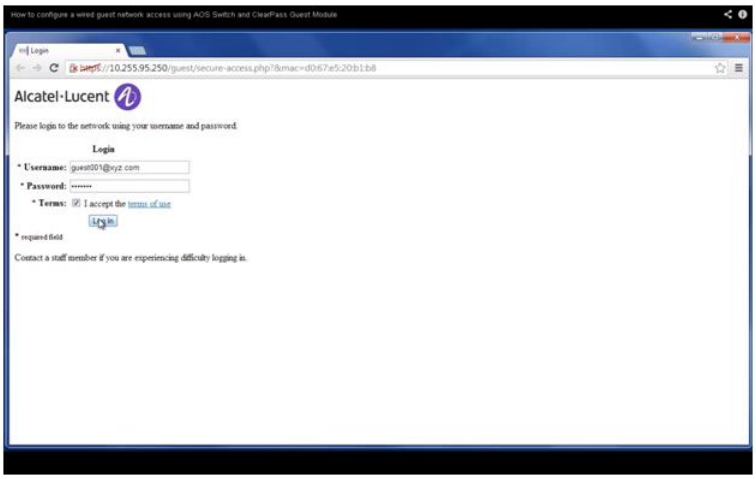
Step 2: ClearPass (Guest) - Create Profiles

Create Restricted Profile: **Enforcement->Profiles** **Template:** RADIUS Based Enforcement **Name:** ALU Restricted Profile **Type:** RADIUS **Action:** Accept **Attribute Type:** Radius:IETF, Alcatel-Lucent-Enterprise **Attribute Name:** Filter-ID, Alcatel-Redirection-URL **Attribute Value:** UNP-restricted, (redirect URL)	
Create Guest Profile: **Enforcement->Profiles** **Template:** RADIUS Change of Authorization (CoA) **Name:** ALU Guest CoA Profile **RADIUS CoA Template:** Aruba-Change-User-Role **Attributes Type:** Radius:IETF **Attribute Name:** Filter-ID **Attribute Value:** UNP-guest	

Step 3: ClearPass (Guest) - Create MAC and Web Authentication Services

Add MAC Authentication Service **Configuration->Services** **Type:** MAC **Authentication** **Name:** ALU Wired MAC Authentication Service **Monitor Mode:** Disabled **Service Rule Type:** Radius:IETF **Service Rule Name:** NAS-Port-Type **Service Rule Operator:** BELONGS_TO **Service Rule Value:** Ethernet (15) **Authentication Methods:** Allow All MAC AUTH **Enforcement Policy:** ALU Wired MAC Enforcement Policy	
Add Web Authentication Service **Configuration->Services** **Type:** Web-based Authentication **Name:** ALU Wired Captive Portal Authentication Service **Authentication Sources:** [Guest user Repository] [Local SQL DB] **Enforcement Policy:** ALU Wired Captive Portal Enforcement Policy	

Step 4: ClearPass (Guest) - Login Example

Example Redirect	
Example login	

Verifying the BYOD Configuration

A summary of the commands used for verifying the BYOD configuration is given here:

show unp global configuration	Displays global BYOD parameter values, such as the redirection server name, the port bounce status, and pause timer value.
show unp user	Displays the status of the new BYOD clients that access the network.

IoT Device Profiling

The growth of Internet of Things (IoT) devices has challenged the network administrators, as mobile devices lack the option to connect using Ethernet, which is the dominant wired access technology. A single user now uses multiple devices to connect to the network. It is a tough task to reliably identify devices and make sure these devices are compliant and enforce required polices on these devices. Gaining visibility into IoT device types is essential for network administrators to build granular access policies to maintain security and quality of service (QoS) for critical enterprise applications.

IoT Device Profiling allows the network administrators to support and manage smart phones, Tablets and other devices connecting to the network.

The IoT Device Profiling uses DHCP FingerPrinting and MAC OUI (MAC Vendors) to identify IoT devices. Apart from collecting the dhcp-option-55 and dhcp-option-60, the OmniSwitch will also collect the HTTP User-Agent from HTTP Get Request and host names from DNS queries. OmniVista will fetch these information from the switch for further profiling of IOT devices. IPv6 is supported in Device Profiling.

The Device Profiling for IPV6 packets is done using the DHCP-Fingerprinting, DNS-Fingerprinting and HTTP-Fingerprinting as it is done for the IPV4 packets.

MAC organizationally-unique identifiers (OUI) allows to recognize and classifying the device by identifying its MAC address.

DHCP FingerPrinting allows to track the devices on the network and block those are not allowed access. It also helps in analyzing the future growth by accessing the trending information.

The IoT Device Profiling allows to:

- identify and categorize various IoT devices connecting to the network.
- identify the IoT devices based on local device signature database.
- collect signature, collect various packet meta data required for IoT device identification.
- profile devices based on identification.
- use built-in UNPs for IoT device categories such as PoE camera, temperature sensor, heart-rate monitor, medical imaging, and so on for the identified device.
- maintain a database of identified IoT device and un-identified IoT devices for qualitative and quantitative analysis.
- classify the identified IoT devices based on UNP of choice.

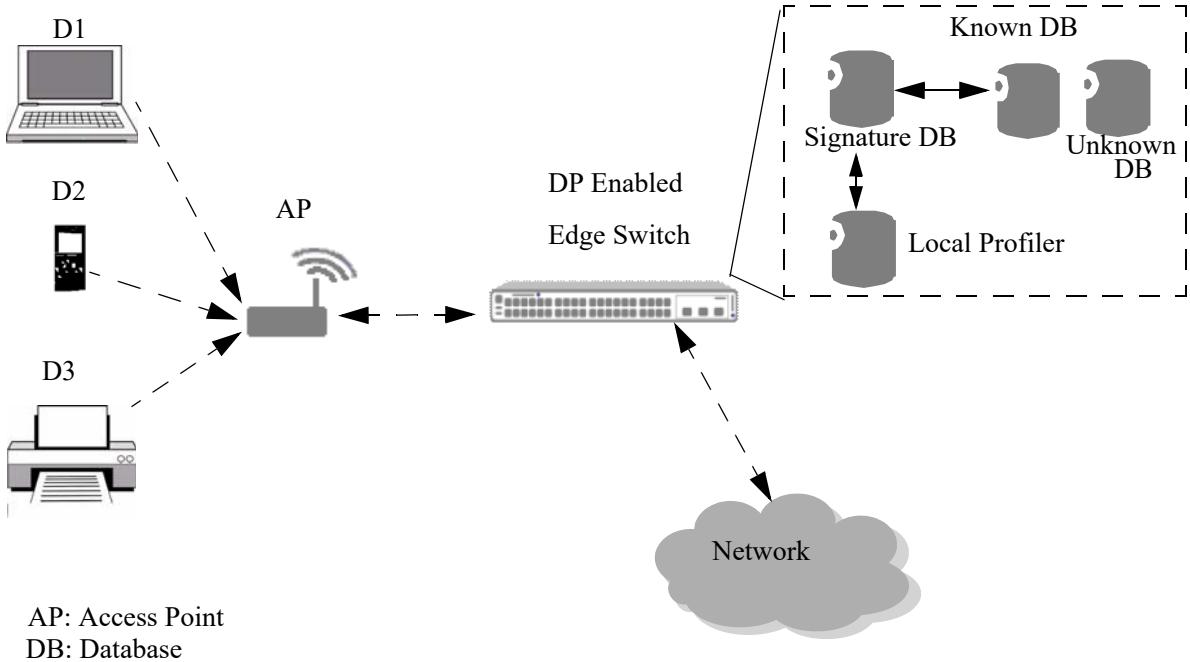
IoT Device Profiling Overview

IoT Device Profiling monitors the devices connecting to the network, detects and profiles the devices at the switch level.

Device Profiling consists of three main components:

- A local signature collector
- A local profiler
- UNP profiling

Consider the following scenario for a basic understanding on how Device Profiling (DP) works:



AP: Access Point
DB: Database
DP: Device Profile

Figure 30-23 : IoT Device Profiling Overview

Consider the device D1 is trying to connect to the network through the access point (AP). The AP is connected to the switch on which DP is enabled. The DHCP packets from the device D1 requesting for IP address from the DHCP server is trapped to the CPU and sent for verification. The DP enabled switch maintains a local signature database for verification of the devices. The received DHCP packets are parsed to collect information such as the DHCP option 55, DHCP option 60 and the MAC address of the device. Once this information is received, the local profiler on the switch will look for the corresponding information in the local signature database.

If the device match is found, a new entry is created in the known database with MAC address, device-type, device-name, port number, initial timestamp and last detected timestamp for reference. Then the UNP profile for the identified device is applied and the required access to the network is provided.

If the device match is not found, the device information is recorded in the unknown database for manual validation and verification.

Quick Steps for Configuring Device Profile

Follow the steps below for configuring Device Profile on the switch:

1 Enable the Device Profiling on the switch by using the command **device-profile admin-state**. For example:

```
-> device-profile admin-state enable
```

2 By default, Device Profiling is enabled on all the ports when it is globally enabled on the switch. To disable the Device Profiling on the specific ports or linkagg on which the client request must not be processed for Device Profiling, use the command **device-profile port linkagg**. For example, the following command disables Device Profiling on linkagg 1 to 5:

```
-> device-profile linkagg 1-5 admin-state disable
```

3 Configure the extended classification rule using the command **unp classification-rule**. For example:

```
-> unp classification-rule ext-r1 precedence 250
```

4 Configure the Device Profiling rule condition for the extended classification rule using the command **unp classification-rule device-type**. For example:

```
-> unp classification-rule ext-r1 device-type Printer
```

Notes:

- If UNP classification is not configured, the IoT device cataloging will work with built-in UNPs for IoT device categories in the local database.
- Device Profiling is enabled by default on UNP ports and is disabled by default on fixed ports.

Device Identification

The IoT Device Profiling uses DHCP FingerPrinting and MAC OUI (MAC Vendors) to identify Internet of Things (IoT) devices.

MAC OUI or MAC Vendors: Every device connecting to the network has its unique identifier. Vendors like Apple and Sony have their own pattern of MAC address. This unique ID is used for identifying the device. The information from the MAC OUI allows to classify unknown devices whose DHCP option 55 is not present in the signature database.

DHCP Fingerprint: During the DHCP protocol exchange, there is an option for the DHCP server to query information on the type of device, manufacturer name, and OS of the client device. It includes a set of DHCP options, such as option 55 and 60. Information in option 55 or 60 is incorporated to form a unique identifier known as the DHCP FingerPrint. For example, the option 55 sequence for an Apple iPhone can be one of the following:

1,3,6,15,119,78,79,95,252

1,3,6,15,119,95,252,44,46,47

DHCP option 60 tracks the vendor ID. For example, for certain Cisco VoIP devices, the vendor ID can be Cisco Systems, Inc. IP Phone, which is very generic; or it can be Cisco Systems, Inc. IP Phone 7912, which is more specific.

This information forms a unique identifier, the DHCP FingerPrint.

Local Database and Signature Management

The Device Profiling consists of signature database, known database, and unknown database.

The Device Profiling network interface (DP-NI) collects the information required for DP. The packets are parsed and meta data is built from the information. The collector will parse only those DHCP packets which are initiated by client, such as DHCP-DISCOVER, DHCP-REQUEST and DHCP-INFORM, for DP. The local profile running on the DP enabled switch will retrieve the meta data from the interprocess communication (IPC) and look for the information in the DHCP option 55 database which is the signature database.

For devices which finds a signature match, a new entry is created in DP in the device database which is the known database. A maximum of 10000 entries can be stored in this database. Once the threshold is reached the first entry is overwritten by new entry.

For devices which does not find a signature match, the device information such as MAC Address, MAC Vendor, DHCP option 55 and DHCP option 60 information, port number, initial timestamp and last detected timestamp is stored in the unknown database. A maximum of 10000 entries can be stored in this database. Once the threshold is reached the first entry is overwritten by new entry.

Adding or Removing a New Device Type

An unknown device type can be added or removed from DP. The devices in the unknown database can be viewed for analysis before adding. To view the unknown device, use the command **show device-profile catalog** with the **unknown** parameter option. For example:

```
-> show device profile catalog unknown

Port/LAGG  Mac-Address      DHCP VCI (Option 60) DHCP Option 55                    Mac-Vendor
---------+-----------------+-------------------+----------------------------------+----------
1/1/15    34:E7:0B:03:C5:B0 HAP.1-OAW-AP1221-US 1,3,6,12,15,28,42,43,66,67,138,212 HANNetwork
1/1/13    34:E7:0B:03:C5:B1 HAP.1-OAW-AP1220-US 1,3,6,12,15,28,42,43,66,67,138,212 HANNetwork
1/1/3     11:32:54:0A:32:54 alcatel.noe.0       1,3,28,43,58,59
0/1       16:56:34:0B:33:21   -                 1,3,6,15,119,252
```

The unknown device can be assigned to a device type and device name, moving it from unknown database to known database.

The unknown devices can be added to the known database from its DHCP option 55 information or its MAC address.

To add the device to known database, use the command **device-profile device-type**. For example:

```
-> device-profile device-type IP-Camera device-name netgear from dhcp-option-55
1,3,6,15,119,252

-> device-profile device-type printer device-name netprint from mac-address
11:32:54:0A:32:54
```

To remove the device, use the **no** form of the command. For example:

```
-> no device-profile device-type IP-Camera
```

Updating the Signature

When a new device is added, the signature database must also be updated with the device signature for future use. Use the command **device-profile update-signature** to update the custom or user-defined device type to the flash and signature database. For example:

```
-> device-profile update-signature
```

Two signature files are maintained in the switch for Device Profiling, one for MAC vendor and other for DHCP Fingerprinting. The signature database can be updated with the new signature file, using the command **device-profile update-signature from**. For example:

```
-> device-profile update-signature from /flash/dhcp_option55_list.txt
```

Automatic UNP Profile Assignment in Device Profiling

When a device gets identified and categorized, the UNP profile can be automatically assigned to the device. To enable automatic assignment of UNP profile, use the command **device-profile auto-unp-assignment**. For example:

```
-> device-profile auto-unp-assignment
```

To disable the automatic assignment of UNP profile, use the **no** form of the command. For example:

```
-> no device-profile auto-unp-assignment
```

Classification Rule for Device Profiling

The precedence value 255 in UNP extended classification rule is reserved for Device Profiling classification rule. The following extended classification rules are automatically defined when Device Profiling is enabled for the switch (these rules cannot be removed):

- devProfPrinter
- devProfWindows
- devProfIP-Phone
- devProfWireless-Router
- devProfSmartPhone/PDA/Tablets

A Device Profile rule condition can be configured for a specific extended classification rule. Use the command **unp classification-rule device-type** to define a DP rule condition. For example:

```
-> unp classification-rule ext-r1 device-type Printer
```

The Device Profile classification rule has the highest precedence in the classification. However, if a device is learned on a profile from RADIUS through 802.1x or MAC-Authentication, the classification rule cannot be applied for the device. The profile configuration returned from RADIUS is retained.

UNP Enforcement of Device Profile

Unified Network Access Policies are a critical feature for IoT device enablement. DHCP Device Fingerprinting can quickly identify IoT devices from multiple manufacturers. Device profiling is implemented by DHCP options that provide vendor-specific information about the device hardware or operating system. The exchange is done using DHCP options such as option 55 and 60. Utilizing DHCP options provides vendor, device, and OS information, which combined constitute the device “fingerprint”.

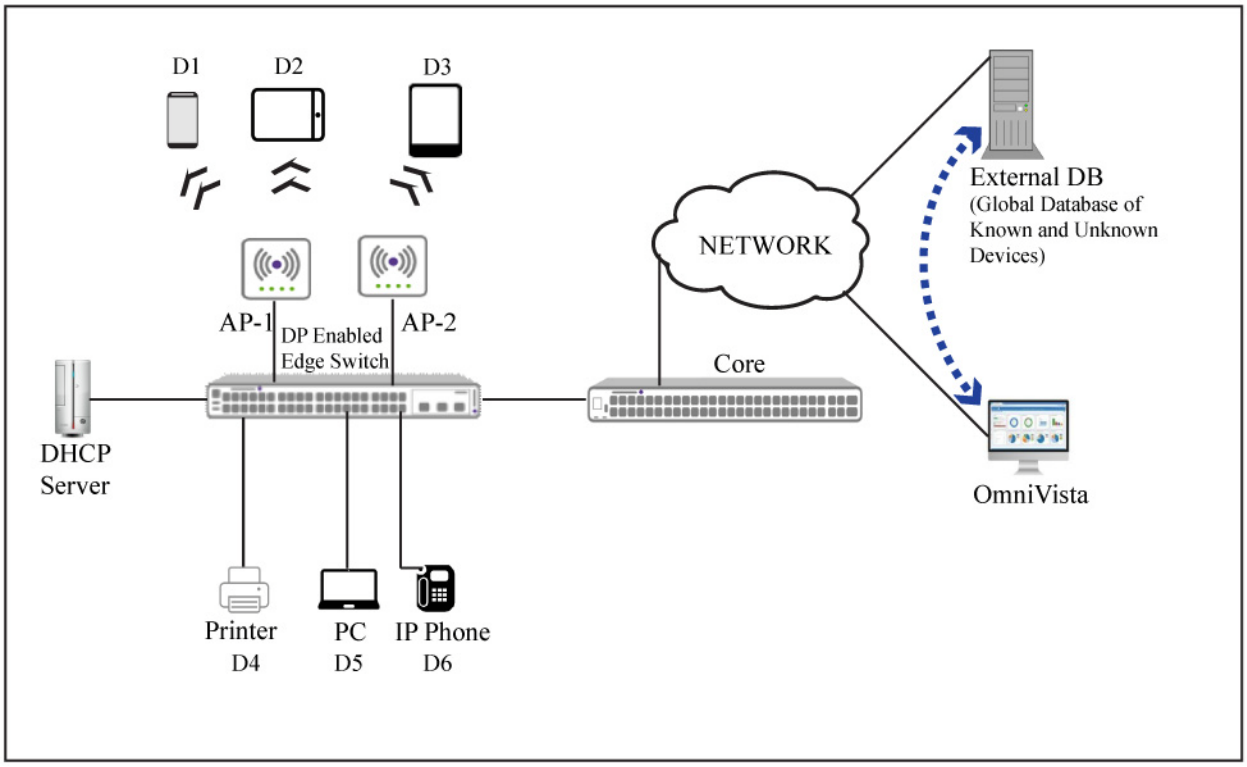


Figure 30-24 : IoT Example - OV interaction with OmniSwitch

In the above example, when an unknown device is connected to the switch, it sends DHCP Discover message that is flooding to the network to get IP address. The DHCP server in the network assigns an IP address to the endpoint device, while the switch assigns a UNP guest profile. The endpoint device connected to the switch gets active after getting assigned with the IP address.

The endpoint device then sends DHCP Option 55, Option 60 and MAC OUI information, apart from HTTP GetRequest and DNS queries to OmniVista (OV), which is in constant communication with the Switch. The OV in turn sends this information to an external fingerprint database server. The external fingerprint database server is a global database of known and unknown devices.

The external fingerprint database server compares the information with the parameters stored in the database. The database server then sends the appropriate type of device information to OV.

OV runs the device profiler engine and the enforcement engine, which triggers a notification back to the switch. OmniSwitch supports an SNMP table with following fields.

- MAC-address [Index of the SNMP table]
- Device-Name
- Device-Type
- UNP-Profile

OV sends the SNMP set notification to the OmniSwitch using the supported SNMP table. The switch holding this MAC address with the UNP (MAC, Category, Device name, Profile) enforce or disapprove the device with the UNP profile.

The Device Profile enabled on the switch on receiving this SNMP set for enforced UNP profile will send a notification to the Access Guardian module to apply the appropriate device profile.

Note. It is required to reset the SNMP set on the OmniSwitch, if any, before the enforcement of UNP profile.

Verifying the Device Profile Configuration

A summary of the commands used for verifying the Device Profile configuration is given here:

show device-profile config	Displays the global configuration of the Device Profile.
show device-profile summary	Displays the number of device identified based on the device type.
show device-profile catalog	Displays the details of known and unknown devices identified in the network.
show device-profile signatures	Displays the content in the signature file.
show device-profile signatures from	Applys the new signature file in the system.

31 Configuring Application Monitoring and Enforcement

Application usage patterns in the enterprise network is changing with the increase in use of the social networking, browser based file sharing, and peer to peer applications. The use of these applications result in the new traffic patterns in the network that are not straightforward to distinguish.

OmniSwitch Application Monitoring and Enforcement (AppMon) feature addresses the key challenges of real time classification of flows at application level by providing differential QoS treatment in the form of higher priority marking and security policies at application level. AppMon feature improves the quality of user experience through application aware network optimization and control.

Application Enforcement feature allows the switch to differentiate between different traffic flows and assign the proper QoS and Security policies. The feature also provides appropriate QoS marking to applications flows as per the Application-aware user configuration QoS policies.

Application Monitoring feature collects and reports the application specific flow information over a period. Based on this data, application specific enforcement policies can be designed.

Note.
- AppMon is supported on a virtual chassis of OmniSwitch 6860 and OmniSwitch 6860E platforms where at least one OmniSwitch 6860E is mandatory for the feature to work.

- AppMon is supported on a virtual chassis of OmniSwitch 6860N.

In This Chapter

This chapter provides an overview of the AppMon feature and describes how to configure this feature through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following information and procedures are included in this chapter:

- "AppMon Defaults" on page 31-3.
- "Application Monitoring and Enforcement Overview" on page 31-4
- "Application Signature File/Kit" on page 31-8
- "Quick Steps for Configuring AppMon" on page 31-7.
- "Configuring AppMon" on page 31-10
- "Configuration Guidelines" on page 31-11
- "Verifying AppMon Configuration" on page 31-21.

AppMon Defaults

Description	Keyword	Default
AppMon default global status	**app-mon admin-state**	Disabled
AppMon default status on the physical port	**app-mon port admin-state**	Disabled
AppMon L3 mode	**app-mon l3-mode**	Enabled for both monitoring and enforcement (both Ipv4, Ipv6)
AppMon L4 mode	**app-mon l4-mode**	Enabled (both TCP and UDP)
AppMon flow-table enforcement statistics	**app-mon flow-table enforcement stats**	Disabled
Flow table aging interval (only Enforcement)	**app-mon aging enforcement**	Application specific
Logging threshold	**app-mon logging-threshold**	20000 flows
AppMon flow sync collection interval (only Enforcement)	**app-mon flow-sync enforcement interval**	60 seconds
Set enforcement to IPv4 or IPv4-IPv6 flows.	**app-mon enforcement-mode**	IPv4-IPv6

Application Monitoring and Enforcement Overview

Application Monitoring and Enforcement (AppMon) feature is provided to address the key challenges of real time classification of flows at application level. It provides differential QoS treatment in the form of higher priority marking and security policies at application level. AppMon feature improves the quality of user experience through application aware network optimization and control.

The following are the key components for AppMon:

- Application Signature Kit File - the file containing application signatures.
- Application Pool - pool of supported applications.
- Application List - list of applications supported for monitoring or enforcement. Applications can be added to this list using the application name or application group.
- Application Group - a group of applications. The group can be added to the application list.
- QoS policy: QoS policy configuration at application level or application group level.

Application Monitoring

The following diagram provides a high-level example of Application Monitoring process:

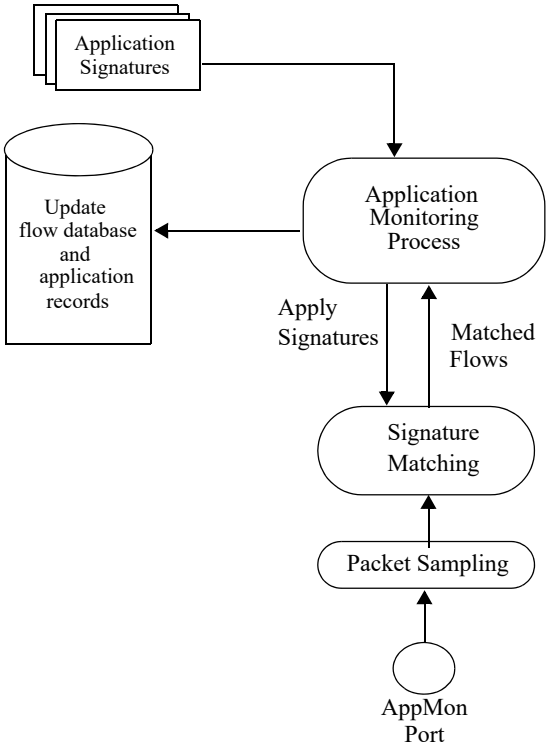


Figure 31-1 : Application Monitoring Process

Application monitoring functionality includes:

- Identify traffic flows from attached networks at Layer 3.
- Application recognition of these traffic flows.
- Signature toolkit upgrade from OmniVista.
- Report the identified application flow information.
- Report application specific flow details including 5-tuple flow credentials, total number of flows, number of flows per application, and so on.

Application Enforcement

The following diagram provides a high-level example of Application Enforcement process:

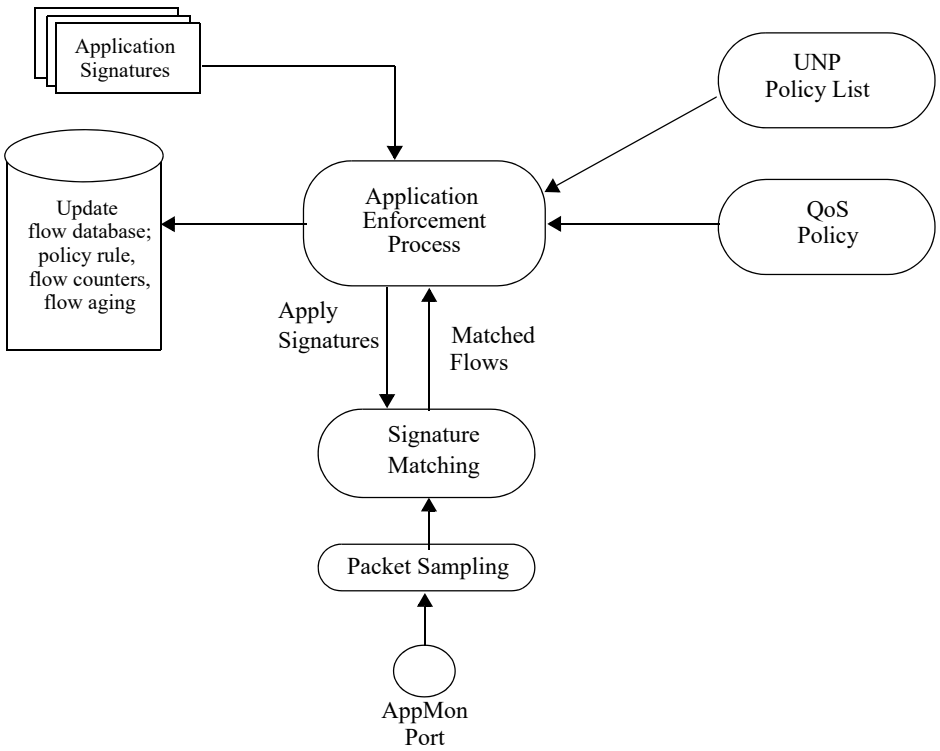


Figure 31-2 : Application Enforcement Process

Application enforcement functionality includes:

- Identify traffic flows from attached networks at Layer 3.
- Application recognition of these traffic flows.
- Signature toolkit upgrade from OmniVista.
- Report the identified application flow information to CPU for UNP/VNP association.
- Policy enforcement that provides user configured application specific QoS treatment for application traffic flows including higher DSCP/802.1p or internal priority marking that translates to higher egress

queue for traffic scheduling. QoS treatment can also include security policies that can perform policy action such as dropping or rate limiting the flows.

- Report application specific flow details including 5-tuple flow credentials, total number of flows, number of flows per application, and so on.
- Associate QoS policies with UNP profile and provide user level policy treatment. Main QoS policy action includes DSCP/802.1p/Priority Marking, Disposition drop/accept, and rate-limiter. Application flows may come with DSCP marking as found appropriate by the end device. Application enforcement feature overrides these values if QoS is configured, and marks them with values customized for network as per configuration.
- Display application information as part of per UNP user level.
- Existing QoS policy configuration framework is used for the enforcement policy configuration where application name/group can be specified. QoS policies provides treatment to bi-direction flow basis and not per uni-direction flow.

Quick Steps for Configuring AppMon

The following quick steps provide a brief tutorial for configuring AppMon on OmniSwitch.

1 Use the **app-mon admin-state** command to globally enable AppMon functionality on the switch.

```
-> app-mon admin-state enable
```

2 Use the **app-mon port admin-state** command to enable AppMon functionality on one or more switch ports. For example, the following command enables AppMon on ports 1/1/2 through 1/1/5:

```
-> app-mon port 1/1/2-5 admin-state enable
```

3 Add or remove applications or application groups to an application list for enforcement or monitoring using the **app-mon app-list** command. Separate application list is maintained for both enforcement and monitor features. For example:

```
-> app-mon app-list enforcement add app-name whatsapp
-> app-mon app-list monitor add app-group apg2
```

4 To enable the set of application signatures configured for AppMon, use the **app-mon apply** command. This activates both enforcement and monitoring application lists for flow classification.

```
-> app-mon apply
```

Note. Verify the AppMon configuration using the **show app-mon config** and **show app-mon port** commands. For example:

```
-> show app-mon config
Admin State                           : Enable,
Operational State                     : Enable,
Separate Config File Status           : Yes,
L3-IPv4                               : Enable,
L3-IPv6                               : Enable,
Enforcement Flow-Table Stats          : Enable,
Enforcement Flow-Sync Interval        : 60 seconds,
Monitor Logging Threshold             : 7000,
Enforcement Logging Threshold         : 7000,
App-Pool Applications                 : 2849,
Monitor Applied Applications          : 2849,
Enforcement Applied Applications      : 2849,
Upgraded Signature File Type          : Production,
AOS Compatible Signature Kit Version  : 1,
Signature Kit version                 : 3.7.6

-> show app-mon port
  Port     Admin-Status   Oper-Status       L4-mode
----------+-------------+------------------+---------------
1/1/1       Enable         Up                TCP-UDP
1/1/2       Enable         Up                TCP-UDP
1/1/3       Enable         Up                TCP-UDP
1/1/4       Enable         Up                TCP-UDP
1/1/5       Enable         Up                TCP-UDP
1/1/6       Enable         Up                TCP-UDP
1/1/7       Enable         Up                TCP-UDP
.
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in this display.

Application Signature File/Kit

Signatures are pattern recipes that are chosen for uniquely identifying an associated application (or protocol). When a new application or protocol is encountered, it is analyzed and an appropriate signature is developed and added to a database (referred to as a signature library).

Signature File Update

OmniSwitch boots up with the Factory Default Signature file. The application pool is created with the factory default kit. Other supported AppMon functionality, such as configuring application groups, configuring auto-application groups, adding to the application list, and so on, are user configurable.

- If there is any new default signature file present in the new AOS image, then AOS replaces the installed old default signature file with the new factory default signature file.
- OmniSwitch boots up with the factory default signature file. When the next signature file update is done through OmniVista (for more information, refer to OmniVista documentation), the new signature file is installed on top of the factory default file provided the required license is available. In case of license expiry or unavailability of license, switch continues to work with the existing signature file installed with valid license.
- On successful installation of the new signature file, the following actions take effect:
  - On every signature update, application pool will be recreated. The newly added applications are seen in the application pool.
  - If the 'auto-group create' option is used, 'category' based application groups will be created.
  - If the application group already exists, then all the auto-application groups will be updated with applications available in the newly updated application pool.
  - In case of application name missing in the new signature file: This will be treated as deleted application from the new application pool. Such applications are removed from the application pool, application group, application list, and active application list.
- Once the production signature file is installed, AOS does not roll back to the factory default signature file.
- AOS compatible signature file version determines the compatibility between AOS software and the Signature file. AOS maintains its own compatibility version and new signature file must have the same compatibility version to successfully update on AOS. This must remain same for both. Switch allows the signature file upgrade only when the compatibility version between AOS and the signature file match.
- In case of application added application name missing in the new signature file: This will be treated as deleted application from the new application pool. Such applications are removed form application pool, application groups, application list, and active application list. The newly added applications are seen in the application pool.

Application Flow Database

When a match occurs between IP traffic and the application signature, a flow entry is created based on the following 5-tuple:

- Source IP Address (IPv4 or IPv6)
- Destination IP Address (IPv4 or IPv6)
- Source L4 port
- Destination L4 port
- IP Protocol (TCP or UDP)

Configuring AppMon

This section provides the following information about how to configure AppMon on the OmniSwitch:

- "Configuration Guidelines" on page 31-11.
- "Enabling/Disabling AppMon" on page 31-13.
- "Enabling/Disabling AppMon Per Port or Slot" on page 31-13
- "Create Auto-Groups" on page 31-14
- "Configuring Application Group" on page 31-14.
- "Configuring Application List" on page 31-15.
- "Activate Applications for AppMon" on page 31-15
- "Configuring L3 Mode of Operation" on page 31-16
- "Configuring L4 Mode of Operation" on page 31-16
- "Configuring Enforcement Mode for IPv4/IPv6 Flows" on page 31-16
- "Clearing Flow Table Entries" on page 31-17
- "Configuring Flow Table Statistics Update" on page 31-17
- "Configuring Aging Interval" on page 31-17
- "Configuring Logging Threshold" on page 31-18
- "Configuring Sync Interval" on page 31-18
- "Configuring Force Flow Sync" on page 31-18
- "Clearing Application List" on page 31-19
- "Configuring AppMon Enforcement QoS Policy Rules" on page 31-19
- "Separate File for AppMon Configuration" on page 31-20

Configuration Guidelines

Review the guidelines in this section before configuring AppMon on the OmniSwitch.

- AppMon works on an application level and not on individual application events/operations. On configuring an application, all associated events are considered for application monitoring and enforcement.
- Supports only IP traffic (TCP or UDP).
- On OmniSwitch 6860, AppMon must not be configured on user ports and uplink ports at the same time.
- On OmniSwitch 6860N, it is mandatory to enable AppMon on user ports and uplink ports. AppMon must be enabled on both ports involving bi-direction flow traffic.
- AppMon does not support link aggregate interface. AppMon is supported at individual port level only. Also, port will not be allowed to be configured in the link aggregate if AppMon is enabled on the port.
- On OmniSwitch 6860N, AppMon can be configured on a port that is part of a link aggregate.
- AppMon configuration is not allowed on Virtual Fabric Link, ERP, VLAN stacking, or port mirroring ports.
- Does not support tunneled traffic, encrypted traffic, and fragmented traffic (supported only if initial fragmented packet contains the signature).
- Software policy lookup considers AppMon enforcement specific policies for a given application name only when it is part of an active application list. In case of policy configured both for application and application group where same application is part, policy will be selected based on what is configured in the active application list. Active application list allows only one application at a time, either directly added in the application list or added through an application group.
- Application enforcement cannot be provided to IP flows which moves between NIs (due to link aggregate, STP block scenario, or L3 ECMP group configuration).
- If an AppMon flow is detected on a UNP port, then AppMon UNP policy list is applied to the flow. If UNP policy list is not configured, then default QoS policy list is applied. For non-UNP ports, default QoS policy list is applied. The **show unp user details** command displays the list of enforcement applications used by the UNP user. For example:

```
-> show unp user details
Port: 4/1/6
 MAC-Address: 00:80:9f:a0:65:94
   Access Timestamp                = 02/18/2014 04:42:33,
   User Name                       = 00:80:9f:a0:65:94,
   IP-Address                      = 25.1.1.25,
   Vlan                            = 25,
   Authentication Type             = Mac,
   Authentication Status           = Authenticated,
   Authentication Failure Reason   = -,
   Authentication Retry Count      = 0,
   Authentication Server IP Used   = 135.254.163.143,
   Authentication Server Used      = cppm,
   Server Reply-Message            = -,
   Profile                         = UNP-device,
   Profile Source                  = Auth - Pass - Server UNP,
   Profile From Auth Server        = UNP-device,
   Classification Profile Rule     = -,
```

```
    Role                            = pl3,
    Role Source                     = L2-Profile,
    User Role Rule                  = -,
    Restricted Access               = No,
    Location Policy Status          = -,
    Time Policy Status              = -,
    Captive-Portal Status           = -,
    QMR Status                      = Passed,
    Redirect Url                    = -,
    SIP Call Type                   = Not in a call,
    SIP Media Type                  = None,
    Applications                    = whatsapp
Total users : 1
```

- For QoS policies, AppMon can be enabled for enforcing policy actions such as drop, 802.1p/DSCP priority marking, and rate limiting. Only the edge switches need to enable AppMon enforcement functionality for higher priority QoS treatment. Intermediate switches in the network must only provide consistent QoS treatment based on the earlier marking done by edge switches, whereas core switches only provide QoS treatment.

Enabling/Disabling AppMon

To enable AppMon feature globally on the switch, use the **app-mon admin-state** command with the **enable** option. By default, AppMon is disabled on the switch.

```
-> app-mon admin-state enable
```

To disable AppMon functionality, use the **app-mon admin-state** command with the **disable** option.

```
-> app-mon admin-state disable
```

Enabling/Disabling AppMon Per Port or Slot

To enable AppMon on one or more switch ports, use the **app-mon port admin-state** command with **enable** option as shown. It is mandatory to enable AppMon globally for the port level AppMon to function.By default, AppMon is disabled on all the switch ports.

For example, the following command enables AppMon on ports 1/1/2 through 1/1/5:

```
-> app-mon port 1/1/2-5 admin-state enable
```

For example, the following command enables AppMon on slot 1.

```
-> app-mon slot 1/1 admin-state enable
```

When slot option is used, then AppMon configuration is applied on all physical ports of that particular slot.

To disable AppMon functionality on a port or slot, use the **app-mon port admin-state** command with **disable** option. For example:

```
-> app-mon port 1/1/2-5 admin-state disable
-> app-mon slot 1/1 admin-state disable
```

Create Auto-Groups

Auto-Group functionality automatically creates application-groups based on the classification of supported applications in the signature file. The 'Category' field is used for classification (for example, Youtube will be part of Web category, and Webex will be part of Audio/Video category).

To create application groups automatically on the switch, use the **app-mon auto-group create** command.

```
-> app-mon auto-group create
```

On using the **app-mon auto-group create** command, the following action will be taken:

- Category-based application groups are created. If the application group already exist with category names, those groups are updated from the existing application pool. Application group update includes addition of all the applications of a given category from the application pool to the corresponding auto-application group. The auto-application group retains any additional applications added by the user.
- Modifications are allowed in auto-application groups with addition of applications using the **app-mon app-group** command. Deletion of applications is also allowed for any application in the group. Modifications to the auto-application groups are retained on reboot, if saved by using the **appapp-mon apply** and **write memory** commands.
- On signature file update, if the "auto-group create" option is given, category-based application groups are created or updated if they already exist.
- It is mandatory to enter **appapp-mon apply** to save the auto-group configuration on the switch.

Configuring Application Group

Applications can be added to an application group. To create an application group, use the **app-mon app-group** command. You can add individual applications or a range of applications to an application group. Only those applications that are part of an application pool are allowed to be added to an application group. This group can be used for enforcement or to monitor features.

To add a specific application to an application group, use the following command:

```
-> app-mon app-group apg2 add app-name whatsapp
```

To add a range of applications or multiple applications to an application group, use the **from** and **to** options.

```
-> app-mon app-group apg1 add from sip to viber
```

To remove an application from an application group, use the following command:

```
-> app-mon app-group apg2 remove app-name whatsapp
```

Configuring Application List

Configure an application list for enforcement or monitoring. The list can be formed with individual applications or with application groups. A separate application list is maintained for both enforcement and monitor features.

To add or remove application or application group to an application list, use the **app-mon app-list** command. You can add individual application or an application group to an application list. The application group can be user created or generated automatically (see "Create Auto-Groups" on page 31-14 and "Configuring Application Group" on page 31-14).

For example, the following command adds an individual application to the application list.

```
-> app-mon app-list add app-name whatsapp
```

For example, the following example adds an application group to the application list. Application group enables addition of multiple applications to the application list.

```
-> app-mon app-list add app-group apg1
```

To remove an application from an application list, use the following command. For example:

```
-> app-mon app-list remove app-name whatsapp
```

Activate Applications for AppMon

To enable the set of application signatures configured for both enforcement and monitoring application lists for flow classification, use the **app-mon apply** command. This command activates the application list configuration. This command saves the current application-list, application-group, and auto-groups configuration to flash when the **write memory** command is used.

```
-> app-mon apply
```

When **app-mon apply** command is used, any application configured more than once in an application list (individually or as a part of application group) is checked. The **app-mon apply** command will not be successful until the conflict is resolved. Use the **show app-mon app-list** command with the **monitor conflict** or the **enforcement conflict** parameter option to display the available conflicts in an application list. The duplicate application names must be removed for successful **app-mon apply**.

Configuring L3 Mode of Operation

To enable or disable monitoring and enforcement for IPv4 flows, IPv6 flows, or both, use the **app-mon l3-mode** command. By default, monitoring and enforcement is enabled for both IPv4 and IPv6 flows.

For example, the following command enables monitoring and enforcement for IPv4 packets:

```
-> app-mon l3-mode ipv4 admin-state enable
```

The following command disables monitoring and enforcement for IPv4 packets:

```
-> app-mon l3-mode ipv4 admin-state disable
```

Configuring L4 Mode of Operation

To enable or disable monitoring and enforcement for TCP or UDP flows, use the **app-mon l4-mode** command. By default, both TCP and UDP flows are processed. For example:

The following command enables monitoring and enforcement for UDP flows:

```
-> app-mon port 1/1/2 l4-mode udp admin-state enable
```

The following command disables monitoring and enforcement for UDP flows:

```
-> app-mon port 1/1/2 l4-mode udp admin-state disable
```

A specific L4 port range can be excluded from the AppMon operation using the **app-mon l4port-exclude** command. The valid Exclude ID range is 1–8. The valid port range is 1–65535.

For example, to exclude TCP port 20 to port 30 from the AppMon operation, use the following command:

```
-> app-mon l4port-exclude range-id 5 tcp-service-port start 20 end 30
```

Use the **no** form of the **app-mon l4port-exclude** command to delete the range. For example,

```
-> no app-mon l4port-exclude range-id 6
```

Configuring Enforcement Mode for IPv4/IPv6 Flows

To enable the enforcement for either IPv4 or IPv4-IPv6 flows, use **app-mon enforcement-mode** command. By default, enforcement mode is set to IPv4-IPv6. Enforcement mode can be changed only when AppMon is in disabled state.

For example, the following command applies enforcement to IPv4-IPv6 flows.

```
-> app-mon enforcement-mode ipv4
-> app-mon enforcement-mode ipv4-ipv6
```

Clearing Flow Table Entries

To clear all the learned flow-table entries from the monitor or enforcement application list, use the **app-mon flow-table flush** command. For example:

```
-> app-mon flow-table enforcement flush
-> app-mon flow-table monitor flush
```

Configuring Flow Table Statistics Update

To enable or disable flow table statistics update for enforcement applications, use the **app-mon flow-table enforcement stats** command. This command is applicable only for enforcement applications.

The statistics collection capability is shared with the Service Manager feature, which means that either Service Manager or AppMon can use this capability at any given time. Hence, disabling the counter usage by Service manager using the **service stats** command is required to view the flow table statistics update for enforcement applications. For more information about this command, see the "Service Manager Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following command enables flow-table statistics update for enforcement applications:

```
-> app-mon flow-table enforcement stats admin-state enable
```

The following command disables flow-table statistics update for enforcement applications:

```
-> app-mon flow-table enforcement stats admin-state disable
```

Configuring Aging Interval

Configures aging time for dynamically learned TCP/UDP flows for each application for Enforcement applications.

Flow aging is supported for the applications that are part of the enforcement application list. Flows related with enforcement application list are made active for QoS treatment as well statistics collection. Flow aging is not supported for applications that are part of Monitor application list. Monitor flow tables log these flows when they are detected until logging threshold is reached.

The aging interval time can be specified in the range of 3–120 minutes. By default, the aging interval is set per application and TCP or UDP flow type basis. TCP and UDP flows generated by a given application age out based on the TCP/UDP configured value.

To configure the aging interval, use the **app-mon aging enforcement** command. For example:

```
-> app-mon aging enforcement app-name sip interval 60m
```

To configure the default aging interval, use the **default** keyword. For example:

```
-> app-mon aging enforcement app-name tftp default
```

Configuring Logging Threshold

To configure the threshold value for the number of matched flows for enforcement and monitor applications, use the **app-mon logging-threshold** command.

The maximum number of flows to be logged is in the range of 1000–60000 flows. When the logging threshold value is set to '0', flows are not logged to the log file.

When used with the monitor option, the threshold is configured for the number of matched flows to be displayed in the monitor flow table commands. For example, the following command configures a threshold value of 10000 for monitor applications.

```
-> app-mon logging-threshold monitor num-of-flows 10000
```

When used with the enforcement option, the threshold is configured for the number of matched flows to be saved on to the log file for enforcement applications. For example, the following command configures a threshold value of 30000 for enforcement applications.

```
-> app-mon logging-threshold enforcement num-of-flows 30000
```

To configure the default logging threshold, 20000, use the **default** keyword as shown.

```
-> app-mon logging-threshold monitor num-of-flows default
```

Configuring Sync Interval

Configures the interval at which the enforcement flows information is refreshed. The range for the interval is 10–3600 seconds. The interval can be configured only for the enforcement feature.

Use the **app-mon flow-sync enforcement interval** command to force the flow sync. For example:

```
-> app-mon flow-sync enforcement interval 10
```

To configure the default interval value, 60 seconds, use the **default** keyword as shown.

```
-> app-mon flow-sync enforcement interval default
```

Default flow-sync interval is 300 seconds for monitoring. This is not user configurable.

Configuring Force Flow Sync

To synchronize the flows learned in the data path in real time, use the **app-mon force-flow-sync** command. By default, flow synchronization occurs every 5 minutes for monitor flows and 60 seconds for enforcement flows.

For example:

```
-> app-mon force-flow-sync enforcement
-> app-mon force-flow-sync monitor
```

Clearing Application List

To remove all the applications from the enforcement or monitor application list, use the **clear app-mon app-list** command. This command does not clear the active application list, until 'app-mon apply' is used.

For example:

```
-> clear app-mon app-list enforcement
-> clear app-mon app-list monitor
```

Configuring AppMon Enforcement QoS Policy Rules

AppMon enforcement allows the switch to differentiate between different traffic flows and assign the proper QoS and Security policies. The following set of policy actions are supported: Disposition Drop/ Accept, Rate limiting, DSCP Marking, 802.1p Marking, Internal Priority Marking, Redirection, and Mirroring.

- QoS policy rules can be configured for a given application as well as an application group where the same application also exists. The QoS policy applied is based on what is configured in the application list at the time of **app-mon apply**.
- QoS policy lookup considers AppMon specific policies for a given application name only when it is part of enforcement active application list. In case the policy is configured both for application name and application group where same application is part, policy will be selected based on what is configured in the enforcement active application list.
- QoS policy lookup is performed for the flow when AppMon signature matched packet is received. If there is any change to existing QoS policies, modified policies are effective only for the new flows. However, the following scenarios are supported:
  - If there is a change in policy action, existing matched flows are updated for new policy action.
  - If AppMon specific policy rules are removed, flows to which the rules were applied are moved to the default policy.
  - On QoS disable, existing flows are moved to the default policy.
  - On QoS flush or apply, existing flows are moved to the default policy.
  - Default policy refers to disposition accept policy action.
- While configuring an AppMon policy rule, multiple condition parameters (for example, source IP, destination IP, source VLAN and so on) cannot be defined in a single policy condition along with AppMon application name or group.
- The application group needs to be created before adding the application group in QoS policies.
- While configuring the QoS policies, the application name or the application group must exist in AppMon.
- Destination port or port group based policy condition is not supported for AppMon.

Configure AppMon enforcement QoS policy rules using the **policy condition** command. An application name or application-group name can be specified in the policy condition as shown below.

```
-> policy condition c1 app-mon-application-group appgroup1
-> policy condition c2 app-mon-application-name watsapp
```

For more information about configuring QoS policy rules for AppMon, see the "QoS Policy Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* and the Chapter 28, "Configuring QoS." chapter in this guide.

Separate File for AppMon Configuration

Use **app-mon separate-config-file** command to reduce per application level AppMon configuration displayed in the "show configuration snapshot" as well as in "vcboot.cfg". By default, separate configuration file.option is enabled.

When this command is used, per application configuration is moved to a separate file called appmon_vcboot.cfg from vcboot.cfg on 'write memory'.

For example, on enabling **app-mon separate-config-file** command, "show configuration snapshot app-monitoring" command displays as follows:

```
-> show configuration snapshot app-monitoring
! APP-MONITORING:
app-mon separate-config-file
app-mon apply app-mon flow-sync enforcement interval 10
app-mon apply
app-mon admin-state enable
app-mon flow-table enforcement stats admin-state enable
```

Specify the **app-snapshot** parameter with the **show app-mon config** command to view the AppMon configuration at the application level.

Verifying AppMon Configuration

A summary of the **show** commands used for verifying the AppMon configuration is given here.

show app-mon config	Displays global AppMon configuration, which includes information about admin-state, running mode, IP mode, aging-timer, and total signatures.
show app-mon port	Displays AppMon status per physical port or per slot for the switch.
show app-mon app-pool	Displays all the applications that are part of an application pool.
show app-mon app-list	Displays a list of applications and application groups added to an application list.
show app-mon app-group	Displays the details of all the applications in an application group.
show app-mon app-record	Displays current-hour application-record information as well the historic application-records on the hourly or 24-hours basis for monitored applications.
show app-mon ipv4-flow-table	Displays the flow table for IPv4 flows entries for enforcement and monitor flows.
show app-mon ipv6-flow-table	Displays the flow table for IPv6 flows entries for enforcement and monitor flows.
show app-mon l4port-exclude	Displays the port range excluded from AppMon operation.
show app-mon stats	Displays the number of flow statistics.
show app-mon aging enforcement	Displays the aging interval for each application for enforcement feature.
show app-mon vc-topology	Displays the AppMon VC topology.

32 Configuring Application Fingerprinting

The OmniSwitch Application Fingerprinting (AFP) feature attempts to detect and identify remote applications by scanning IP packets and comparing the packets to pre-defined bit patterns (application signatures). Once an application is identified, AFP collects and stores information about the application flow in a database on the local switch. Additional configurable options for this feature include the ability to apply QoS policy list rules to the identified flow and generating SNMP traps when a signature match occurs.

Using this implementation of AFP, an administrator can obtain more detailed information about protocols running on a specific device or make sure that certain QoS actions are automatically applied wherever an application might be running.

In This Chapter

This chapter provides an overview of the Application Fingerprinting feature and describes how to configure the port-based functionality and profile attributes through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following information and procedures are included in this chapter:

- "AFP Defaults" on page 32-2.
- "Quick Steps for Configuring AFP" on page 32-4
- "AFP Overview" on page 32-5.
- "Interaction With Other Features" on page 32-9.
- "Configuring AFP" on page 32-10.
- "Verifying the AFP Configuration" on page 32-18.

AFP Defaults

Description	Keyword	Default
The status of AFP functionality on the switch.	**app-fingerprint admin-state**	Enabled
The status of AFP activation on switch ports and link aggregates.	**app-fingerprint port**	Disabled
ASCII text file containing REGEX application signatures.	**app-fingerprint signature-file**	flash/app-signature/ app-regex.txt
Trap generation when IP traffic flow matches an application signature	**app-fingerprint trap**	Disabled

Default REGEX Application Signatures

The default "app-signature.txt" file provided in the "/flash/app-signature/" directory on the switch contains the following sample REGEX application signatures and groups (see "Using the Application REGEX Signature File" on page 32-7 for more information):

Application REGEX Signatures
App Name: bgp Description: Border Gateway Protocol \xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff..?\x01[\x03\x04]
App Name: ciscovpn Description: VPN client software to a Cisco VPN server \x01\xf4\x01\xf4
App Name: citrix Description: Citrix ICA - proprietary remote desktop application \x32\x26\x85\x92\x58
App Name: dhcp Description: Dynamic Host Configuration Protocol [\x01\x02][\x01-]\x06.*c\x82sc
App Name: hotline Description: Hotline - P2P filesharing protocol, TRTPHOTL\x01\x02
App Name: jabber Description: open instant messenger protocol <stream:stream[\x09-\x0d][-~]*[\x09-\x0d]xmlns=['"]jabber
App Name: rdp Description: Remote Desktop Protocol, rdpdr.*cliprdr.*rdpsnd
App Name: rtsp Description: Real Time Streaming Protocol, rtsp/1.0 200 ok
App Name: sip Description: Session Initiation Protocol, (invite|register|cancel|message|subscribe|notify) sip[\x09-\x0d -~]*sip/[0-2]\. [0-9]
App Name: smb Description: Samba - Server Message Block, \xffsmb[\x72\x25]

App Name: smtp Description: Simple Mail Transfer Protocol, 220[\x09-\x0d -~]* (e?smtp|simple mail)
App Name: ssh Description: Secure Shell, ssh-[12]\.[0-9]
App Name: vnc Description: Virtual Network Computing, rfb 00[1-9]\.00[0-9]\x0a$
Application Groups
App Group: chatting = jabber
App Group: mail = smtp
App Group: network = bgp dhcp rtsp smb
App Group: p2p = hotline
App Group: remote_access = ciscovpn citrix rdp ssh vnc
App Group: voip = sip

Quick Steps for Configuring AFP

The following quick steps provide a brief tutorial for configuring Application Fingerprinting to monitor and profile host applications on the network:

1 Use the **app-fingerprint admin-state** command to globally enable Application Fingerprinting functionality on the switch:

```
-> app-fingerprint admin-state enable
```

2 Use the **app-fingerprint port** command to enable AFP functionality on one or more switch ports or link aggregates. Once enabled, IP traffic received on the port is sampled and compared to application REGEX signatures that reside in an ASCII text file on the local switch. For example, the following command enables AFP on port 1/1/23 to monitor and identify IP packets that match the REGEX signatures in the "my-p2p" application group:

```
-> app-fingerprint port 1/1/23 monitor-app-group my-p2p
```

Monitoring is one of three operational modes supported on AFP ports. See "Application Fingerprinting Modes" on page 32-6 for more information.

3 *Optional*. By default, the "app-regex.txt" file located in the "/flash/app-signature/" directory on the switch contains the REGEX signatures to which IP flows are compared. To specify a different filename for the signatures, use the **app-fingerprint signature-file** command. For example:

```
-> app-fingerprint signature-file app2_regex.txt
```

4 *Optional*. Use the **app-fingerprint reload-signature-file** command to load the contents of a new or updated application signature file into switch memory. For example:

```
-> app-fingerprint reload-signature-file
```

5 *Optional*. Use the **app-fingerprint trap** command to enable trap generation when an IP flow matches an application signature. For example:

```
-> app-fingerprint trap enable
```

Note. *Optional*. Verify the Application Fingerprinting configuration using the **show app-fingerprint configuration** and **show app-fingerprint port** commands. For example:

```
-> show app-fingerprint configuration
Admin-state:          Enabled,
SNMP Trap:            Enabled,
Signature File:       app-regex.txt

-> show app-fingerprint port
Legend: * = Port or App-Group is invalid

Port    Operation Mode          App-group/Policy-list
--------+---------------+----------------------------------------
1/1/23    Monitoring        my-p2p
1/1/24    QoS               list1
1/1/25    Unp               UNP
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in this display.

AFP Overview

The OmniSwitch Application Fingerprinting (AFP) feature attempts to detect and identify remote applications by scanning IP packets received on an AFP port and comparing the packet contents against predefined bit patterns or signatures. Once the application is identified, the switch can collect the source and destination information, apply QoS, or generate an SNMP Trap.

This feature is best utilized on server-facing ports where an administrator needs more detailed information on the protocols running inside a device, or to make sure that certain QoS actions are automatically applied wherever an application might run. By default, Application Fingerprinting is globally enabled for the switch, but disabled on all switch ports and link aggregates.

Enabling Application Fingerprinting on a port or link aggregate triggers the sampling of IP packets on that port or aggregate.

- The sampled IP packets are compared against REGEX application signatures stored in the default “app-regex.txt” file located in the “/flash/app-signature/” directory on the local switch.
- When an application REGEX match occurs, the switch generates a multi-tuple classifier based on information obtained from the matching application packets. This classifier is then stored into a local database on the switch and is used to identify the application for further monitoring or action.
- After the application is identified on a specific port, Application Fingerprinting monitors the application traffic flow on that port. In addition, the switch may also apply QoS policy rules to the application traffic flow, depending on the Application Fingerprint port configuration.

The following diagram provides a high-level example of the Application Fingerprinting process:

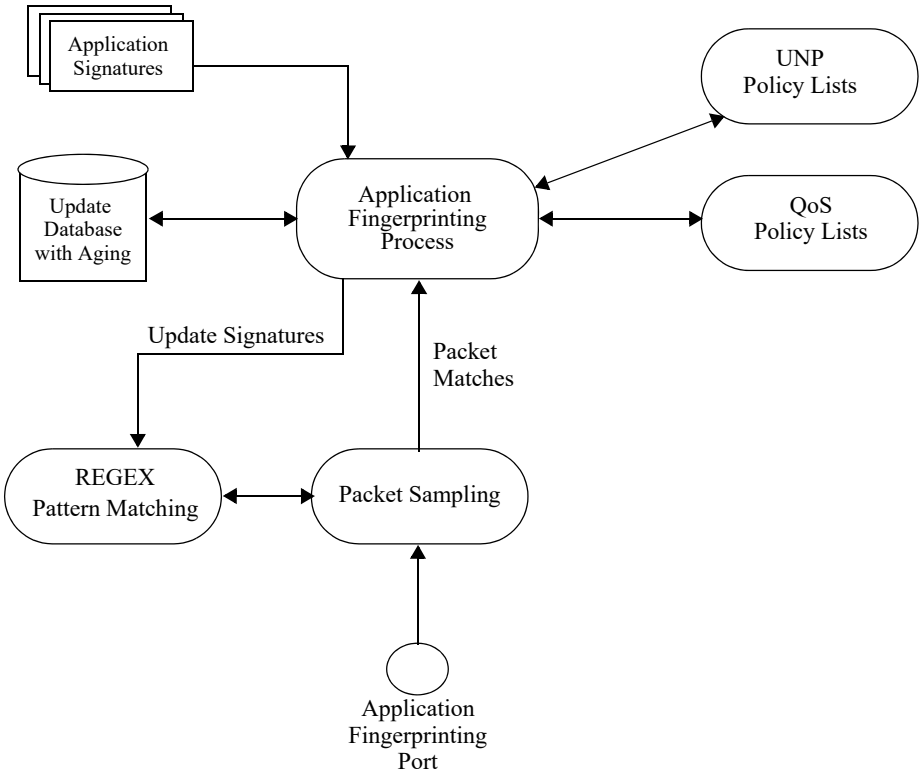


Figure 32-1 : AFP Overview

Application Fingerprinting Modes

The Application Fingerprinting process is enabled on a per-port basis. When configuring a port or link aggregate as an AFP port, the user must also specify one of three operational modes for the port: monitoring, QoS, or UNP.

All three of these modes will monitor ingress traffic on the AFP port to detect any IP packets that match REGEX signatures. When a match occurs identifying information is scanned from the packets and logged into a local database on the switch. However, the three modes differ when it comes to determining which group of REGEX signatures to monitor and if any QoS actions are applied to the matching traffic.

REGEX signatures can be grouped into an application group; the selected AFP mode specifies which application group to monitor (see "Using the Application REGEX Signature File" on page 32-7). QoS policies are applied through policy lists associated with the AFP port or through lists associated with a Universal Network Profile (UNP).

Note. Configuring more than one operating mode type for the same port is allowed, but using a different application group for each mode configured on the port is highly recommended. One advantage to using different groups for different modes on the same port is that you can have one group of applications that are just monitored and another group of applications to which QoS is applied.

Using the Monitoring Mode

When a port is configured to operate in AFP monitoring mode, the name of an application group of signatures is specified. This triggers the switch to sample ingress IP packets on that port and compare the packets to the signatures in the specified application group. After an application is identified and logged into the local database, no further action is taken and monitoring of the matching traffic continues.

The monitoring mode is particularly useful to initially identify and monitor remote applications entering the network. The administrator can use the information gathered during monitoring to determine if any subsequent QoS actions are needed.

Using the QoS Mode

Using the QoS mode is similar to using the monitoring mode in that both modes trigger the sampling of IP packets on the port. The difference is that configuring QoS mode specifies a QoS policy list name instead of an application group name. The policy list specifies the application group to monitor.

The policy list assigned to the AFP port must contain a policy rule with a policy condition that specifies the name of an application group to monitor. The rule can also contain policy actions to apply to the matching application traffic.

The **appfp-group** policy condition and **appfp** policy list type are used to configure QoS policies for matching application traffic.The following is an example QoS policy rule and policy list configuration that is associated with an AFP port that is configured to run in the QoS mode:

```
-> policy condition c1 appfp-group my-p2p
-> policy action a1 disposition drop
-> policy rule r1 condition c1 action a1 no default-list
-> policy list drop_my-p2p type appfp
-> policy list drop_my-p2p rule r1

-> app-fingerprint port 1/2/5 policy-list-name drop-p2p
```

Using the UNP Mode

Using the Universal Network Profile (UNP) mode also triggers IP packet sampling on the port but first attempts to see if the ingress traffic is classified into a UNP.

- If the traffic is assigned to a UNP, the switch then checks if the UNP is associated with an AFP QoS policy list that contains the AFP policy condition.
- If the UNP is associated with an AFP QoS policy list, the application group specified in the AFP policy condition of a rule within that list is used to monitor ingress traffic on the AFP port. Policy actions associated with the same AFP policy condition rule are applied to matching IP traffic.
- If there is no matching UNP or the UNP does not use an AFP policy list or condition, then AFP ignores the traffic; no packet sampling or monitoring is performed.

The UNP QoS policy list for AFP is created in the same manner as how the list used by the QoS mode is created. The main difference between the UNP and QoS mode is the check for UNP classification before packet sampling and monitoring is started. In addition the policy list type is set to UNP instead of Application Fingerprinting and UNP is enabled on the AFP port. For example, the following QoS policy rule and policy list configuration is associated with a UNP that is applied to AFP port traffic associated with the UNP:

```
-> policy condition c1 appfp-group p2p
-> policy action a1 disposition drop
-> policy rule r1 condition c1 action a1 no default-list
-> policy list list1 type unp
-> policy list list1 rules r1
-> qos apply

-> unp profile V10_1 qos_policy-list list1
-> unp profile v10_1 map vlan 10
-> unp classification mac-address 00:00:00:00:03:01 vlan-tag 10 profile1 V10_1
-> unp port 1/2/1 port-type bridge

-> app-fingerprint port 1/2/1 unp-profile
```

Using the Application REGEX Signature File

The REGEX signatures that AFP uses to detect and monitor remote applications are stored in an ASCII text file named “app-regex.txt”. This file is located in the “/flash/app-signature/” directory on the local switch, and the contents of the file is user-configurable.

The application REGEX signature file contains two sections: one section to define application signatures and the other section to define application groups.

- The application signatures section defines a name, optional description, and a REGEX signature for each application.
- The application group section is used to group application signatures together. Each group is identified by a name and consists of the names of each application signature that is a member of the group.

An application group name is required when configuring an AFP port to run in monitoring mode and when creating QoS policy lists that are used when the port is running in the QoS or UNP mode (see “Application Fingerprinting Modes” on page 32-6). Combining multiple application signatures into one group eases configuration; specifying a single group name when configuring the AFP operation requires less steps than having to configure AFP for each individual application.

The “app-regex.txt” file contains a sample configuration to use as a guide for defining AFP application signatures and groups (see “AFP Defaults” on page 32-2).

Application Fingerprinting Database

When a match occurs between an IP traffic flow and a REGEX application signature, the following multi-tuple classifier is generated and stored in a local switch database to identify and track the application associated with the flow:

- Ingress Port
- Dest MAC
- Src MAC
- VLAN
- Dest IP
- Src IP
- Dest Port
- Src Port

Each database entry is subject to a 15 minute aging period. If the database fills up, older entries are aged out before the 15 minute limit (fast aging). However, fast aging is not applied to database entries associated with QoS. In this case, the QoS is removed after the regular 15 minute aging time period expires. When a database entry is removed due to regular aging or fast aging conditions, any corresponding QoS is also removed for that flow.

In addition, a packet counter for each application on the ingress Application Fingerprinting port is kept for statistics generation. The database classification entries and statistics are displayed using Application Fingerprinting **show** commands (see “Verifying the AFP Configuration” on page 32-18).

Interaction With Other Features

This section contains important information about how Application Fingerprinting (AFP) functionality interacts with other OmniSwitch features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

General

- IPv4 and IPv6 packets are sampled on AFP ports. The entire packet is scanned, not just the payload.
- Fragmented, encrypted, control or protocol packets (for example, ICMP, LLDP, BPDU) are not supported.
- AFP is applied after other OmniSwitch features, such as Universal Network Profile (UNP), Learned Port Security (LPS), QoS, DHCP, and other protocols.

QoS

- AFP shares QoS system resources with other OmniSwitch applications. As a result, AFP functionality is subject to available QoS system resources, especially when the QoS and/or UNP modes are running on AFP ports.
- A QoS policy list is used by the AFP QoS and UNP modes to specify the name of an application group of signatures to apply to AFP port traffic.
  - When using the QoS mode, the policy list must be configured as an AFP list (**appfp**) when the list is created. With UNP mode the policy list is not configured as an AFP list, since it associated with a UNP.
  - The QoS **appfp-group** policy condition is used to specify the name of the AFP application group to apply. If this condition is not used in a policy list rule, then no QoS is applied to the AFP port traffic.

sFLOW

AFP uses the OmniSwitch SFLOW mechanism to sample the IPv4 and IPv6 packets.

- The packet sampling rate is approximately 50K packets per second per NI module. This rate may change based on the level of use of switch resources by other applications.
- The sFLOW mechanism runs two seconds for each port (one port at a time), so if there are two AFP ports in an NI, each port is serviced for half of its testing duration, whether or not the port receives any number of packets.
- Do not run AFP and other sFLOW-based services on the same port.

Configuring AFP

This section provides the following information about how to configure and activate the OmniSwitch implementation of Application Fingerprinting:

- "Configuration Guidelines" on page 32-10.
- "Enabling/Disabling AFP" on page 32-11.
- "Enabling/Disabling Trap Generation" on page 32-11
- "Changing the REGEX Signature Filename" on page 32-12.
- "Defining Application REGEX Signatures and Groups" on page 32-13.
- "Configuring AFP Port Modes" on page 32-16.

Configuration Guidelines

Review the guidelines in this section before attempting to configure and activate OmniSwitch Application Fingerprinting (AFP).

- The AFP pattern matching function compares IP packets to REGEX signatures. These signatures are defined in a user-configurable ASCII text file located on the switch. This file also defines groups of application signatures and associates each group with a specific name. A group name is required when configuring AFP functionality (operational modes) on a switch port or link aggregate. Make sure the appropriate application groups are defined in the text file.
- Configuring different operational modes (monitoring, QoS, or UNP) on the same AFP port is allowed, but use different application groups for each mode to avoid conflicts or inconsistencies in how traffic is processed. For example, if monitoring mode is set to use application group named "appgroup1", then configure QoS mode on that same port to use a policy list that specifies application group name "appgroup2".
- QoS, UNP, and AFP QoS use shared switch resources. So configuring an AFP port to run in the QoS mode, UNP mode, or both modes, will require additional QoS resources that may be limited depending on what else is running on the switch.
- Make sure a QoS policy list assigned directly to an AFP port running in the QoS mode is configured as an Application Fingerprinting policy list type (**appfp**). In addition, the policy list rules must contain the **appfp-group** policy condition that specifies the name of an application signature group that AFP uses for packet pattern matching. For example:

```
-> policy condition c1 appfp-group myp2p
-> policy action a1 disposition drop
-> policy rule drop-p2p condition c1 action a1 no default list
-> policy list afp-p2p type appfp
-> policy list afp-p2p rules drop-p2p
-> app-fingerprint port 1/1/5 policy-list-name afp-p2p
```

- A QoS policy list assigned to a UNP that is associated with an AFP port running in the UNP mode must also contain policy list rules that use the **appfp-group** policy condition to specify the application group used for packet pattern matching. However, configuring the policy list as an AFP list type is not required when the list is associated with a UNP.

- If a policy list assigned to an AFP port or assigned to a UNP associated with an AFP port does not contain a rule with the **appfp-group** condition, sampled IP traffic on the port is not matched against any REGEX signatures to determine if any QoS actions in the rule are applied to that traffic.
- The following list of QoS policy actions are supported in policy rules that use the Application Fingerprinting policy conditions:
  - maximum bandwidth
  - disposition drop/accept
  - priority
  - tos
  - 802.1p
  - dscp

Enabling/Disabling AFP

By default, the AFP feature is globally enabled for the switch. To disable this feature, use the **app-fingerprint admin-state** command with the **disable** option. For example:

```
-> app-fingerprint admin-state disable
```

Disabling this feature stops the AFP process but does not remove any AFP port settings or the REGEX signature text file from the switch.

To enable AFP functionality, use the **app-fingerprint admin-state** command with the **enable** option. For example:

```
-> app-fingerprint admin-state enable
```

When globally enabled, the AFP process is triggered only on AFP ports and link aggregates. See "Configuring AFP Port Modes" on page 32-16 for information about how to configure AFP ports.

Verifying the Global AFP Status

Use the **show app-fingerprint configuration** command to verify the global AFP status for the switch. For example:

```
-> show app-fingerprint configuration
Admin-state:          Enabled,
SNMP Trap:            Disabled,
Signature File:       app-regex.txt
```

Enabling/Disabling Trap Generation

Trap generation can occur whenever AFP detects a match between IP packets and application group signatures. Use the **app-fingerprint trap** command to globally enable or disable trap generation for the switch. For example:

```
-> app-fingerprint trap enable
-> app-fingerprint trap disable
```

By default, AFP trap generation is disabled for the switch.

Verifying the Trap Generation Status

Use the **show app-fingerprint configuration** command to verify the trap generation status. For example:

```
-> show app-fingerprint configuration
Admin-state:          Enabled,
SNMP Trap:            Disabled,
Signature File:       app-regex.txt
```

Changing the REGEX Signature Filename

A default REGEX signature file, named “app-regex.txt” is provided in the “/flash/app-signature/” directory on the OmniSwitch. This file is a user-configurable ASCII text file. Adding, removing, or changing application signatures and groups defined in this file is allowed. It is also possible to use a completely different signature file instead of the default “app-regex.txt” file.

To direct AFP to use a different REGEX signature file for pattern matching, make sure the file is uploaded to the “/flash/app-signature/” directory on the switch, then use the **app-fingerprint signature-file** command to point AFP to the new filename. For example, the following command configures AFP to use the “net-regex.txt” file:

```
-> app-fingerprint signature-file net-regex.txt
```

Verifying the REGEX Signature Filename

The **show app-fingerprint configuration** command displays the name of the REGEX signature filename that AFP is using for pattern matching. For example:

```
-> show app-fingerprint configuration
Admin-state:          Enabled,
SNMP Trap:            Disabled,
Signature File:       app-regex.txt
```

Reloading the REGEX Signature File

The REGEX signature file is loaded into switch memory where the contents of the file is accessed by the AFP pattern matching function. If a new filename is designated for the REGEX signature file or if the contents of the current signature file is changed, a reload of that file into the switch memory is required.

Changes to the signature file contents or filename are applied when the signature file is reloaded into memory. A switch reboot is not required.

The **app-fingerprint reload-signature-file** command is used to reload the current signature file. For example:

```
-> app-fingerprint reload-file
```

Note that the current signature file reloaded is either the “app-regex.txt” file, unless a different signature filename was designated as the current signature file using the **app-fingerprint signature-file** command.

Defining Application REGEX Signatures and Groups

To define a new application signature entry in the REGEX signature file, use the following formatting conventions:

`App-name:` *application-name*
`Description:` *application-description*
REGEX-signature

Application signature formatting guidelines:

- The application signature “Description:” field is optional, but the “App-name:” field and REGEX signature are required.
- Maximum characters allowed for the “App-name:” field is 24.
- Maximum characters allowed for “Description:” field is 64.
- REGEX signature guidelines:
  - Signature should be more than 6 characters but less than 256 characters.
  - Do not start the signature with the “^” character (because scanning starts at the beginning of the packet not from the beginning of the packet payload).
  - The “.”, “*”, “?”, or any combination of the three characters may not work properly on hex value data in the packet payload (for example, the .* may not work properly).
  - The . * ? represents any single character except carriage return (0xD) and tabs (0x9, 0xB) and may not work with other non-character hex values.
  - Use /x hex notation when possible (for example, instead of "yahoo.com" use "yahoo/x2Ecom" in the signature - ASCII value for . is 0x2E).
  - Be careful about using a space (white space) in the signature.
  - Do not use very complex set of REGEX notation, instead, break it down to multiple simple REGEX signatures.

To define a new signature group in the REGEX signature file, use the following formatting conventions:

`App-group:` *app-group-name* `=` *application-name-1 application-name-2 application-name-3 . . .*

Application signature formatting guidelines:

- Maximum characters allowed for the “App-group:” field is 24.
- Enter a list of application signature names (already defined in the signature file) after the “=” with a space between each name

Verifying the Application Signature and Group Definitions

Use the **show app-fingerprint app-name** and **show app-fingerprint app-group** commands to display the application signature and group configuration defined in the REGEX signature file. For example:

```
-> show app-fingerprint app-name

App Name:        hotline
  Description:     Hotline - P2P filesharing protocol,
  REGEX signature: TRTPHOTL\x01\x02

App Name:        jabber
  Description:     open instant messenger protocol,
  REGEX signature: <stream:stream[\x09-\x0d ][ -~]*[\x09-\x0d ]xmlns=['"]jabber

App Name:        sip
  Description:     Session Initiation Protocol,
  REGEX signature: (invite|register|cancel|message|subscribe|notify) sip[\x09-
\x0d -~]*sip/[0-2]\.[0-9]

App Name:        smtp
  Description:     Simple Mail Transfer Protocol,
  REGEX signature: 220[\x09-\x0d -~]* (e?smtp|simple mail)

-> show app-fingerprint app-group

App Group:  chatting
  App names:  jabber

App Group:  mail
  App names: smtp

App-group: network
  App names: bgp dhcp rtsp smb

App Group:  p2p
  App names:  hotline

App-group: remote_access
  App names: ciscovpn citrix rdp ssh vnc

App Group:  voip
  App names: sip
```

Example REGEX Signature File

This section contains an example “app-regex.txt” file. Note that application signatures and groups are defined using the formatting conventions described in “Defining Application REGEX Signatures and Groups” on page 32-13.

```
App-name: TCP-Syn-BDos
Description: TCP-Syn-BDos
\x02\xfe..\x80.*\xc0\xa8\x05\xca.*(\x0c|\x04)\x00\x00\x50

App-name: UDP-Flood
Description: UDP-Flood
\x2a.*\xc0\xa8\x05\xca.*\x7a\x69\x00\x87

App-name: DNS-Attack
Description: DNS-Attack
\xc0\xa8\x05\xca.*\x01\x00.*example\x04fake

App-name: Apache-mod_cache-DoS
Description: Apache-Headers-mod_cache-DoS
Cache\x2dControl: +(max\x2dage\x3d|s\x2dmax-
age\x3d|max\x2dstale\x3d|max\x2dage\x3d|min\x2dfresh\x3d)

App-name: BO-Multicast
Description: BO-Borland-StarTe-Multicast
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

App-name: HTTP-Hp-OpVw-OvAccep
Description: HTTP-Misc-Hp-OpVw-OvAccep-BO
OvAcceptLang\x3den\x2dusaAAAAAAAAAAAAAAAAAAAA

App-name: HTTP-null-byte
Description: HTTP-Misc-asp-null-byte-dis-3
/6fNY7wiRTr/VhR9aOCw5WKprcOxYFD57s1kDpoCCekW0Sxhywdx.*wcanQ.*wcanQ


App-group: Static = Apache-mod_cache-DoS BO-Multicast HTTP-null-byte HTTP-Hp-OpVw-OvAccep

App-group: AttackMon = TCP-Syn-BDos

App-group: AttackBlock = UDP-Flood

App-group: AttackRateLmt = DNS-Attack
```

Configuring AFP Port Modes

Configuring a port or link aggregate as an AFP port also applies an operational mode to the port. The operational mode (monitoring, QoS, or Universal Network Profile) applied determines the following:

- The application signature group to use for monitoring ingress IP traffic on the port (monitoring mode).
- The QoS policy list that specifies the application signature group to monitor and any QoS actions to apply to the matching IP traffic (QoS mode).
- Whether or not to check for a Universal Network Profile (UNP) associated with ingress port traffic. If port traffic is associated with a UNP, the QoS policy list associated with the UNP is used to determine the application signature group to monitor (UNP mode).

It is possible to configure more than one operational mode per AFP port. However, using a different application signature group for each mode is highly recommended to avoid conflicts that might cause undesired dropping of traffic, especially when the QoS or UNP modes are both used on the same port.

Configuring the AFP Monitoring Mode

To configure a port or link aggregate as an AFP port operating in the monitoring mode, use the **app-fingerprint port** command with the **monitor-app-group** parameter. For example:

```
-> app-fingerprint port 1/2/1 monitor-app-group my-p2p
-> app-fingerprint linkagg 10 monitor-app-group my-p2p
```

In this example, port 1/2/1 and aggregate 10 are configured as AFP ports that will pattern match and monitor ingress IP packets using the REGEX signatures defined in the “my-p2p” application group. When a match is found, no further action is taken on the matching packets other than logging and monitoring the application traffic.

Configuring the AFP QoS Mode

To configure a port or link aggregate as an AFP port operating in the QoS mode, use the **app-fingerprint port** command with the **policy-list-name** parameter. For example:

```
-> app-fingerprint port 1/2/5 policy-list-name drop-p2p
-> app-fingerprint linkagg 2 policy-list-name drop-p2p
```

In this example, port 1/2/5 and aggregate 2 are configured as AFP ports that will pattern match ingress IP packets using REGEX signatures defined in an application group that is specified by an AFP policy condition in the “drop-p2p” policy list. When a match is found, QoS actions associated with the AFP condition rule are applied to the matching traffic.

Configuring the AFP UNP Mode

To configure a port or link aggregate as an AFP port operating in the UNP mode, configure the port as a UNP port then use the **app-fingerprint port** command with the **unp** parameter. For example:

```
-> unp port 1/1/8 port-type bridge
-> app-fingerprint port 1/1/8 unp-profile
-> unp linkagg 5 port-type bridge
-> app-fingerprint linkagg 5 unp-profile
```

In this example, port 1/1/8 and aggregate 5 are configured as UNP and AFP ports. AFP will determine if traffic received on this port and aggregate is associated with a UNP. If so, the QoS policy list associated with the UNP is applied to the ingress IP traffic. If the policy list does not specify an application group condition, then the AFP port traffic is ignored.

Verifying the AFP Mode Configuration

The QoS and UNP modes both specify application groups for AFP processing. This difference is that the QoS mode directly associates a policy list name with the AFP port, but the UNP mode uses a policy list assigned to a UNP associated with traffic received on the AFP port. The monitoring mode does just that, monitors application group traffic, but does not apply any QoS.

To verify the AFP port configuration for the switch, use the **show app-fingerprint port** command. For example:

```
-> show app-fingerprint port
Legend: * = Port or App-Group is invalid

  Port    Operation Mode          App-group/Policy-list
--------+----------------+---------------------------------------
1/2/1     Monitoring        Testing13
1/2/1     QoS               list1
1/2/1     QoS               list2

-> show app-fingerprint linkagg
Legend: * = Port or App-Group is invalid

  Port    Operation Mode          App-group/Policy-list
--------+----------------+---------------------------------------
0/100     Monitoring        Testing16
0/100     QoS               list3
0/100     QoS               list4
```

Verifying the AFP Configuration

A summary of the **show** commands used for verifying the AFP configuration is given here. For some examples of these commands, see "Quick Steps for Configuring AFP" on page 32-4 and "Configuring AFP" on page 32-10.

show app-fingerprint configuration	Displays the global AFP configuration for the switch. This includes the global status of the feature and trap generation and the name of the REGEX signature file currently in use.
show app-fingerprint port	Displays the AFP port configuration for the switch, including the operational mode and application group applied to the port.
show app-fingerprint app-name	The application signatures defined in the REGEX signature file.
show app-fingerprint app-group	Displays the application groups defined in the REGEX signature file.
show app-fingerprint database	Displays the database entries generated by AFP to identify applications detected on AFP ports.
show app-fingerprint statistics	Displays packet count statistics for the number of packets sampled and packets matched. Also includes the application signature name for the matched packet counts.

33 Managing Authentication Servers

This chapter describes authentication servers and how they are used with the switch. The types of servers described include Remote Authentication Dial-In User Service (RADIUS), Lightweight Directory Access Protocol (LDAP), Terminal Access Controller Access Control System (TACACS+), and SecurID ACE/ Server.

In This Chapter

The chapter includes some information about attributes that must be configured on the servers, but it primarily addresses configuring the switch through the Command Line Interface (CLI) to communicate with the servers to retrieve authentication information about users.

Configuration procedures described include:

- **Configuring a RADIUS Server**. This procedure is described in "RADIUS Servers" on page 33-7.
- **Configuring a TACACS+ Server**. This procedure is described in "TACACS+ Server" on page 33-20.
- **Configuring an LDAP Server**. This procedure is described in "LDAP Servers" on page 33-23.
- **Configuring Kerberos Snooping**. This procedure is described in "Kerberos Snooping Overview" on page 33-40.

For information about using servers for authenticating users to manage the switch, see the "Managing Switch Security" chapter in the *OmniSwitch AOS Release 8 Switch Management Guide*.

For information about configuring RADIUS/ClearPass servers for Alcatel-Lucent's BYOD solution, see Chapter 30, "Configuring Access Guardian."

Server Defaults

The defaults for authentication server configuration on the switch are listed in the tables in the next sections.

RADIUS Authentication Servers

Defaults for the **aaa radius-server** command are as follows:

Description	Keyword	Default
Number of retries on the server before the switch tries a backup server	**retransmit**	3
Timeout for server replies to authentication requests	**timeout**	2
UDP destination port for authentication	**auth-port**	1645*
UDP destination port for accounting	**acct-port**	1646*
The port number for the server	**port**	1812 (SSL disabled) 2083 (SSL enabled)
Whether a Secure Socket Layer is configured for the server	**ssl | no ssl**	**no ssl**

* The port defaults are based on the older RADIUS standards; some servers are set up with port numbers based on the newer standards (ports 1812 and 1813, respectively).

TACACS+ Authentication Servers

Defaults for the **aaa tacacs+-server** command are as follows:

Description	Keyword	Default
Timeout for server replies to authentication requests	**timeout**	2
The port number for the server	**port**	49

LDAP Authentication Servers

Defaults for the **aaa ldap-server** command are as follows:

Description	Keyword	Default
The port number for the server	**port**	389 (SSL disabled) 636 (SSL enabled)
Number of retries on the server before the switch tries a backup server	**retransmit**	3

Description	Keyword	Default
Timeout for server replies to authentication requests	**timeout**	2
Whether a Secure Socket Layer is configured for the server	**ssl | no ssl**	**no ssl**

Quick Steps For Configuring Authentication Servers

1 For RADIUS, TACACS+, or LDAP servers, configure user attribute information on the servers. See "RADIUS Servers" on page 33-7, "TACACS+ Server" on page 33-20, and "LDAP Servers" on page 33-23.

2 Use the **aaa radius-server**, **aaa tacacs+-server**, and/or the **aaa ldap-server** command to configure the authentication server(s). For example:

```
-> aaa radius-server rad1 host 10.10.2.1 10.10.3.5 key amadeus

-> aaa tacacs+-server tac3 host 10.10.4.2 key otna timeout 10
-> aaa ldap-server ldap2 host 10.10.3.4 dn cn=manager password tpub base c=us
```

Note. (*Optional*) Verify the server configuration by entering the **show aaa server** command. For example:

```
-> show aaa server
Server name = rad1
   Server type              = RADIUS,
   IP Address 1             = 10.10.2.1,
   IP Address 2             = 10.10.3.5
   Retry number             = 3,
   Timeout (in sec)         = 2,
   Authentication port      = 1645,
   Accounting port          = 1646
Server name = ldap2
   Server type              = LDAP,
   IP Address 1             = 10.10.3.4,
   Port                     = 389,
   Domain name              = cn=manager,
   Search base              = c=us,
   Retry number             = 3,
   Timeout (in sec)         = 2,
Server name = Tacacs1
   ServerIp                 = 1.1.1.1
   ServerPort               = 49
   Encryption               = MD5
   Timeout                  = 5 seconds
   Status                     = UP
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in this display.

---

3 If you are using ACE/Server, there is no required switch configuration; however, you must FTP the **sdconf.rec** file from the server to the **/network** directory of the switch.

4 Configure authentication on the switch. This step is described in other chapters. For a quick overview of using the configured authentication servers with Authenticated Switch Access, see the *OmniSwitch AOS Release 8 Switch Management Guide*.

Server Overview

Authentication servers are sometimes referred to as AAA servers (authentication, authorization, and accounting). These servers are used for storing information about users who want to manage the switch (Authenticated Switch Access) and users who need access to a particular VLAN or VLANs.

RADIUS, TACACS +, or LDAP servers can be used for Authenticated Switch Access. Only RADIUS servers are supported for Port-based Network Access Control (Access Guardian).

The following table describes how each type of server can be used with the switch:

Server Type	Authenticated Switch Access	Port-Based Network Access Control
RADIUS	yes (except SNMP)	yes
TACACS+	yes (including SNMP)	no
LDAP	yes (including SNMP)	no

Backup Authentication Servers

Each RADIUS, TACACS+, and LDAP server can have one backup host (of the same type) configured through the **aaa radius-server**, **aaa tacacs+-server**, and **aaa ldap-server** commands, respectively. In addition, each authentication method (Authenticated Switch Access, Authenticated VLANs, or 802.1X) can specify a list of backup authentication servers that includes servers of different types (if supported on the feature).

The switch uses the first available authentication server to attempt to authenticate users. If user information is not found on the first available server, the authentication attempts fails.

Authenticated Switch Access

When RADIUS, TACACS+, and/or LDAP servers are set up for Authenticated Switch Access, the switch polls the server for user login information. The switch also polls the server for privilege information (authorization) if it has been configured on the server; otherwise, the local user database is polled for the privileges.

Additional servers can be configured as backups for RADIUS, TACACS+, and LDAP servers.

A RADIUS server supporting the challenge and response mechanism as defined in RADIUS RFC 2865 can access an ACE/Server for authentication purposes. The ACE/Server is then used for user authentication, and the RADIUS server is used for user authorization.

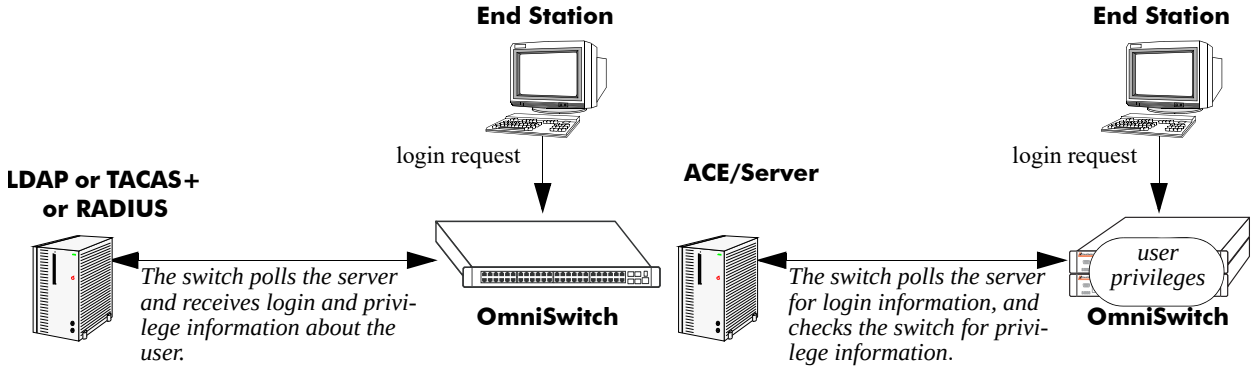


Figure 33-1 : Servers Used for Authenticated Switch Access

RADIUS Servers

RADIUS is a standard authentication and accounting protocol defined in RFC 2865 and RFC 2866. A built-in RADIUS client is available in the switch. A RADIUS server that supports Vendor Specific Attributes (VSAs) is required. The Alcatel-Lucent Enterprise attributes can include VLAN information, time-of-day, or slot/port restrictions.

RADIUS Server Attributes

RADIUS servers and RADIUS accounting servers are configured with particular attributes defined in RFC 2138, RFC 2139, and RFC 3162 respectively. These attributes carry specific authentication, authorization, and configuration details about RADIUS requests to and replies from the server. This section describes the attributes and how to configure them on the server.

Standard Attributes

The following tables list the RADIUS server attributes that are supported by the Alcatel-Lucent RADIUS client. Attribute 26 is for vendor-specific information and is discussed in "Vendor-Specific Attributes for RADIUS" on page 33-10. Attributes used for RADIUS accounting servers are listed in "RADIUS Accounting Server Attributes" on page 33-8.

Num.	Standard Attribute	Notes
1	**User-Name**	Used in Access-Request and Accounting-Request packets.
2	**User-Password**	—
4	**NAS-IP-Address**	Sent with every Access-Request and Accounting-Request. Specifies the source IP address from which packets are sent to the RADIUS server. More than one of these attributes is allowed per user. Configurable through the **aaa radius nas-ip-address** command.
5	**NAS-Port**	Sent with Access-Request and Accounting-Request packets. Slot/port information is supplied in attribute 26 (vendor-specific).
6	**Service-Type**	Framed-User (2) if authentication request type is: - supplicant/802.1x authentication - captive-portal authentication - ASA authentication Call-Check (10) if authentication request type is: - MAC based authentication
8	**Framed-IP-Address**	Sent with every Accounting-Request. Set to the IP address of the user.
11	**Filter-Id**	Sent with every Accounting-Request. Specifies the Universal Network Profile (UNP) name to which the user is classified.
12	**Framed-MTU**	Sent with Access-Request packets. Set to "1400".
18	**Reply-Message**	Multiple reply messages are supported, but the length of all the reply messages returned in one Access-Accept or Access-Reject packet cannot exceed 256 characters.
24	**State**	Sent in challenge/response packets.

Num.	Standard Attribute	Notes
25	**Class**	Used to pass information from the server to the client and passed unchanged to the accounting server as part of the Accounting-Request packet.
26	**Vendor-Specific**	See "Vendor-Specific Attributes for RADIUS" on page 33-10.
27	**Session-Timeout**	Sent with Access-Accept packets. Used to determine the 802.1X re-authentication interval and the session timeout for MAC and Captive Portal users. Configurable through the **aaa session-timeout** command.
29	**Termination-Action**	Used with the Session-Timeout attribute.
30	**Called-Station-Id**	Sent with Access-Request and Accounting-Request packets. Set with the base MAC address of the switch with the configured delimiter (no delimiter by default).
31	**Calling-Station-Id**	Sent with Access-Request and Accounting-Request packets. Set with the MAC address of the user with the configured delimiter (no delimiter by default).
32	**NAS-Identifier**	Sent with every Access-Request and Accounting-Request. Set to the system name of the switch by default. Configurable through the **aaa radius nas-identifier** command.
61	**NAS-Port-Type**	Sent in Access-Request and Accounting-Request packets. Set to "Ethernet (15)".
85	**Acct-Interim-Interval**	Sent in Access-Accept messages. The amount of time, in seconds, between each interim accounting update. Configurable through the **aaa interim-interval** command.
87	**NAS-Port-ID**	Sent in Access-Request and Accounting-Request packets. Set to the user port by default. Configurable through the **aaa radius nas-port-id** command.
95	**NAS-IPv6-Address**	Sent with Access-Request packets. Specifies the IPv6 address of the NAS that is requesting authentication for a user.

RADIUS Accounting Server Attributes

The following table lists the standard attributes supported for RADIUS accounting servers. The attributes in the **radius.ini** file can be modified if necessary.

Num.	Standard Attribute	Description
1	**User-Name**	Used in Access-Request and Account-Request packets.
4	**NAS-IP-Address**	Sent with every Access-Request and Accounting-Request. Specifies the source IP address from which packets are sent to the RADIUS server. More than one of these attributes is allowed per user.
5	**NAS-Port**	Sent with Access-Request and Accounting-Request packets. Slot/port information is supplied in attribute 26 (vendor-specific).

Num.	Standard Attribute	Description
25	**Class**	Used to pass information from the server to the client and passed unchanged to the accounting server as part of the Accounting-Request packet.
32	**NAS-Identifier**	Sent with every Access-Request and Accounting-Request. Set to the system name of the switch by default. Configurable through the **aaa radius nas-identifier** command.
40	**Acct-Status-Type**	Four values must be included in the dictionary file: 1 (acct-start), 2 (acct-stop), 6 (failure), and 7 (acct-on). Start and stop correspond to login/logout. The accounting-on message is sent when the RADIUS client is started. This attribute also includes an accounting-off value, which is not supported.
41	**Acct-Delay-Time**	Set to the amount of time, in seconds, during which the switch is trying to send an Accounting-Request when no Accounting-Response was received.
42	**Acct-Input-Octets**	For 802.1X, MAC, and Captive-Portal users, set to the port counters.
43	**Acct-Output-Octets**	For 802.1X, MAC, and Captive-Portal users, set to the port counters.
44	**Acct-Session-Id**	Unique accounting ID. (For 802.1X, MAC, and Captive Portal users, Alcatel-Lucent uses the MAC address of the client and the login stamp.)
45	**Acct-Authentic**	Indicates how the client is authenticated. Set to RADIUS (1).
46	**Acct-Session-Time**	The elapsed time, in seconds, since the user was authenticated.
47	**Acct-Input-Packets**	For 802.1X, MAC, and Captive-Portal users, set to the port counters.
48	**Acct-Output-Packets**	For 802.1X, MAC, and Captive-Portal users, set to the port counters.
49	**Acct-Terminate-Cause**	Indicates how the session was terminated: NAS-ERROR USER-ERROR LOST CARRIER USER-REQUEST ADMIN-RESET
52	**Acct-Input-Gigawords**	Indicates the number of times Acct-Input-Octets counter has wrapped the 2^{32} (4GB) traffic over the course of the service being provided. This attribute is present in Accounting-Request records where the Acct-Status-Type is set to ‘Stop’ or ‘Interim-Update’.
53	**Acct-Output-Gigawords**	Indicates the number of times Acct-Output-Octets counter has wrapped the 2^{32} (4GB) traffic in the course of delivering the service. This attribute is present in Accounting-Request records where the Acct-Status-Type is set to ‘Stop’ or ‘Interim-Update’.
55	**Event-Timestamp**	Set to the epoch time of transmission for the Account-Request message.

Num.	Standard Attribute	Description
61	**NAS-Port-Type**	Sent in Access-Request and Accounting-Request packets. Set to “Ethernet (15)”.
81	**Tunnel-Private-Group-ID**	Set to the actual VLAN ID to which a user is assigned. This attribute is present in Accounting-Request records where the Acct-Status-Type is set to “Start” or “Stop”.

Vendor-Specific Attributes for RADIUS

The Alcatel-Lucent Enterprise RADIUS client supports attribute 26, which includes a vendor ID and some additional sub-attributes called subtypes. The vendor ID and the subtypes collectively are called Vendor Specific Attributes (VSAs). Alcatel-Lucent Enterprise, through partnering arrangements, has included these VSAs in some vendors’ RADIUS server configurations.

The attribute subtypes are defined in the dictionary file of the server. If you are using single authority mode, the first VSA subtype, Alcatel-Lucent-Auth-Vlan, must be defined on the server for each authenticated VLAN. Alcatel-Lucent Enterprise’s vendor ID is 800 (SMI Network Management Private Enterprise Code).

The following are VSAs for RADIUS servers:

Num.	RADIUS VSA	Type	Description
3	**Alcatel-Lucent-Time-of-Day**	**string**	The time of day valid for the user to authenticate.
4	**Alcatel-Lucent-Client-IP-Addr**	**address**	The IP address used for Telnet only.
6	**Alcatel-Lucent-Port-Desc**	**string**	Description of the port. This attribute is currently defined in the Alcatel dictionary as: • RADIUS attribute type = 26 (VSA) • VSA Vendor ID = 800 • VSA Type = 26 • VSA format = string This attribute is included in all RADIUS messages sent by Alcatel-Lucent OmniSwitch (Access-Request, Accounting-Request Start, Accounting-Request Interim and Accounting-Request Stop).The attribute is set with the alias configured for the port. When the alias is not set, VSA will be an empty string.
9	**Alcatel-Lucent-Asa-Access**	**string**	Specifies that the user has access to the switch. The only valid value is **all**.
39	**Alcatel-Lucent-Acce-Priv-F-R1**	**hex**	Configures functional read privileges for the user.
40	**Alcatel-Lucent-Acce-Priv-F-R2**	**hex**	Configures functional read privileges for the user.
41	**Alcatel-Lucent-Acce-Priv-F-W1**	**hex**	Configures functional write privileges for the user.

Num.	RADIUS VSA	Type	Description
42	Alcatel-Lucent-Acce-Priv-F-W2	hex	Configures functional write privileges for the user.
100	Alcatel-Access-Policy-List	string	• For 802.1X and MAC authenticated users, this attribute overwrites the initial role that is applied based on the policy list associated with the assigned UNP. • For Captive-Portal authenticated users, this attribute assigns a post-login role for the user.
101	Alcatel-Redirect-URL	string	Configures ClearPass to send redirection URL as part of RADIUS response redirecting the user Web traffic.

NAS IP Address Support in RADIUS Packets

NAS IP address is the attribute field in the RADIUS packets which is used to identify the switch that sends the RADIUS packets and used for the accounting process.

The RADIUS client can be configured to include the NAS IP address attribute value in all the outgoing authentication and accounting packets.

In the event where there are multiple switches (NAS) connected to the RADIUS server, all the NAS will have the same source IP address in the IP header. This makes it difficult during authentication and accounting sessions. Configuring this option to include the local NAS IP address in the outgoing RAIUS packets, helps to clearly identify the source.

But in switches managed by OmniVista Cirrus, the interface through which the RADIUS server will be reached is the “VPN IP Address”.

The NAS IP address attribute value can be configured globally or for an AAA profile.

To configure the NAS IP address attribute value for an AAA profile, use the **aaa profile** command. For example:

```
-> aaa profile abc radius nas-ip-address default

-> aaa profile abc radius nas-ip-address local-ip
-> aaa profile abc radius nas-ip-address local-ip 192.168.1.1
```

To configure the NAS IP address attribute value globally, use the **aaa radius nas-ip-address** command. For example:

```
-> aaa radius nas-ip-address default

-> aaa radius nas-ip-address local-ip
-> aaa radius nas-ip-address local-ip 12.12.12.12
```

When the NAS IP address attribute value is configured as “default”, the NAS IP address attribute value is set as the source IP address of the interface used to send the RADIUS packet. In OmniVista Cirrus it will be the VPN IP address.

When the NAS IP address attribute value is configured for a local IP without an IP address, then the NAS IP address attribute value will be the DHCP-Client interface IP address.

When the NAS IP address attribute value is configured for a local IP with an IP address, then this configured IP address value will be used in the NAS IP address attribute.

To display the global AAA attribute values, use the **show aaa radius config** command. For example:

```
-> show aaa radius config
RADIUS client attributes:
  NAS port id         = default,
  NAS identifier      = default
  NAS IP address      = default
    MAC format delimiter:
      Username            = none, UserNameCase = uppercase,
      Password            = none, PasswordCase = uppercase,
      calling station id  = none, ClgStaIdCase = uppercase,
      called station id   = none, CldStaIdCase = uppercase
Unp Profile Precedence = Filter-Id
```

Configuring the MAC Address Format for RADIUS Attributes

A MAC address is used in the User-Name, Password, Calling-Station-ID, and Called-Station-ID attributes. The format of that address is configurable to specify a delimiter to separate the octets within the MAC address and whether the address characters are formatted in upper or lower case.

Consider the following when configuring the MAC address format:

- The MAC address format configured for the User-Name and User-Password attributes is only applied for MAC authentication and accounting, where these attributes are set to the MAC address of the user. The configured format is not applied for 802.1X or Captive Portal authentication and accounting.
- The MAC address format configured for the Called-Station-Id and Calling-Station-Id attributes is applied for MAC, 802.1X, and Captive Portal authentication and accounting sessions when these attributes are set to a MAC address value.
- The Called-Station-Id attribute is set to the base MAC address of the switch.
- The Calling-Station-ID attribute is configurable and can be set to the MAC address or IP address of the user.

By default, the MAC address format does not contain a delimiter (a space is used between octets) and the characters are in uppercase. To change the MAC address format, use the **aaa radius mac-format** command. For example:

```
-> aaa radius mac-format username delimiter ":" case lowercase

-> aaa radius mac-format password delimiter ":" case lowercase
-> aaa radius mac-format calling-station-id delimiter ":" case lowercase
-> aaa radius mac-format called-station-id delimiter ":" case lowercase
```

Use the **show aaa radius config** command to display the current MAC address format in use for each attribute.

Configuring the RADIUS Client

Use the **aaa radius-server** command to configure RADIUS parameters on the switch.

RADIUS server keywords	
host	**auth-port**
key	**acct-port**
hash-key	**vrf-name**
prompt-key	**ssl**
salt	**no ssl**
hash-salt	
retransmit	
timeout	

When creating a new server, at least one host name or IP address (IPv4 or IPv6) (specified by the **host** keyword) is required as well as the shared secret (specified by the **key** or **hash-key** keyword).

In this example, the server name is **rad1**, the host address is 10.10.2.1, the backup address is 10.10.3.5, and the shared secret is **amadeus**. Note that the shared secret must be configured exactly the same as on the server.

```
-> aaa radius-server rad1 host 10.10.2.1 10.10.3.5 key amadeus
```

An option **prompt-key** is provided, which can be used to enter the secret key in a obscured format rather than as clear text. When this option is selected, press the Enter key. A prompt appears prompting to enter the secret key. Secret key needs to be re-entered, and only if both the entries match, command is accepted. Key provided in this mode is not displayed on the CLI as text.

For example,

```
-> aaa radius-server rad1 prompt-key host 10.10.2.1
Enter Key: *******
Confirm Key: *******
```

Salt and hash-salt option are provided to add randomness for the encryption of key.

Use the **salt** option to add randomness to the encryption of key. The maximum length of the salt is 15 characters, and must be in clear text format. By default, system time (24-hour value format) will be taken as default salt value. The user configured or default salt along with the server name will be combined with 'key' and encrypted as a whole, the output of which will be displayed under ‘hash-key'.

```
-> aaa radius-server “Rad1” host 10.10.10.2 key myorg salt mysalt
```

Note. To use a special character in the salt value, put the special character between double quotes ("").

Use the **hash-salt** option to enter the salt value in an encrypted format. The maximum length of the hash-salt is 64 characters.

```
-> aaa radius-server “Rad1” host 10.10.2.1 key myorg hash-salt
c7f5eee2c0f9b33e72e3482673fb6059
```

To modify a RADIUS server, enter the server name and the desired parameter to be modified.

```
-> aaa radius-server rad1 key mozart
```

If you are modifying the server and have just entered the **aaa radius-server** command to create or modify the server, you can use command prefix recognition. For example:

```
-> aaa radius-server rad1 retransmit 5

-> timeout 5
```

For information about server defaults, see "Server Defaults" on page 33-2.

To remove a RADIUS server, use the **no** form of the command:

```
-> no aaa radius-server rad1
```

Note that only one server can be deleted at a time.

RADIUS over TLS

Radius over Transport Layer Security (TLS) provides secured communication between RADIUS and TCP peers using TLS. RADIUS uses MD5 algorithm for secured communication, implementation of TLS further reduces the risk of attack on MD5 encrypted RADIUS packets. There by all RADIUS requests and RADIUS responses are encrypted and transferred between OmniSwitch and RADIUS server.

Configuring TLS for RADIUS Server

To configure TLS for RADIUS server, Secure Sockets Layer (SSL) must be enabled for RADIUS server. To enable SSL use the **aaa radius-server** CLI command.

```
-> aaa radius-server radsrv1 host rad1_ipaddr key rad1_secret vrf-name rad_vrf
ssl
```

To verify the status of SSL for RADIUS server, use the **show aaa server** CLI command. If the SSL enabled is TRUE, then the TLS is enabled for the RADIUS server.

Note. The supported TLS versions are TLSv1.1 and TLSv1.2.

RADIUS Health Check

The RADIUS Health Check feature is used to determine the reachability of a RADIUS server by polling the server at regular time intervals (instead of checking each server in sequence) to determine if the server is up or down. Notification of the server status is then provided to help expedite the authentication process.

Configuring RADIUS Health Check

Configuring the following RADIUS Health Check functionality is done on a per-server basis:

- The status of RADIUS Health Check for the server (enabled or disabled).
- The polling interval—the amount of time after which a health check request is sent to the server.
- The user name and password to use in polling requests to the server.
- Whether or not to re-authenticate users assigned to the authentication server down profile when the RADIUS server comes back up before the authentication server down timeout expires.

By default, RADIUS Health Check is disabled. To enable this functionality for a specific RADIUS server, use **aaa radius-server health-check** command. For example:

```
-> aaa radius-server rad1 health-check
```

Enabling RADIUS Health Check on all RADIUS servers that are configured to provide 802.1X and MAC authentication is recommended. This will help to improve the time it takes to authenticate users.

To disable a RADIUS Health Check session for a specific server. use the **no** form of the **aaa radius-server health-check** command. For example:

```
-> no aaa radius-server rad1 health-check
```

The polling time interval is set to 60 seconds by default. To change the polling interval time, use the **aaa radius-server health-check** command with the **polling-interval** parameter. For example:

```
-> aaa radius-server rad1 health-check polling-interval 300
```

The user name and password that is sent in polling requests to the server are both set to “alcatel” by default. To change these values, use the **aaa radius-server health-check** command with the **username** and **password** parameters. For example:

```
-> aaa radius-server rad1 health-check username admin password switch
```

When the RADIUS authentication server is down (unreachable), user devices are typically assigned to a UNP authentication server down profile. When notification is received that the server is back up, re-authentication of such devices is attempted when the authentication server down timeout value expires.

The RADIUS Health Check functionality provides a failover option that triggers re-authentication without waiting for the authentication server down timeout value to expire. When the health check session receives notification that a server has transitioned from down to up, a re-authentication attempt is immediately triggered for devices that are assigned to the authentication server down profile.

By default, the failover option is disabled (no immediate re-authentication attempt). To enable an immediate re-authentication attempt, use the **aaa radius-server health-check** command with the **failover** parameter. For example:

```
-> aaa radius-server rad1 health-check failover
```

To disable the failover option, use the **no** form of the **aaa radius-server health-check** command with the **failover** parameter. For example:

Use the **show aaa radius health-check-config** command to verify the RADIUS Health Check configuration.

RADIUS Server Statistics

Statistics collection for each RADIUS server defined for the switch is available to analyze transactions between the switch and the RADIUS server. The following transaction information is collected to help an administrator identify network issues and RADIUS server health:

- RADIUS server information and health check status
- Authorization statistics
- Authentication statistics
- Accounting statistics
- Bring Your Own Device (BYOD) statistics

In addition to statistics gathered for RADIUS servers defined for authorization, authentication, and accounting, the Round Trip Time (RTT) for transactions is also calculated. The RTT is the amount of time taken for a single Access-Request/Response transaction (the time difference between when a request is sent to when a response is received). The displayed RTT value is derived from data stored over the last seven days.

RADIUS Server Information and Status

Use the **show aaa server** command to help determine the reachability of a RADIUS server. Information such as server status and whether or not RADIUS Health Check is enabled for the server is displayed. For example:

```
-> show aaa server rad1
Server name = rad1
   Server type          = RADIUS,
   IP Address 1         = 10.10.2.1,
   IP Address 2         = 10.10.3.5,
   Retry number         = 3,
   Timeout (sec)        = 2,
   Authentication port  = 1645,
   Accounting port      = 1646
   SSL enable           = TRUE,
   VRF                  = default
Health Check            = ENABLED,
Primary Server:
           Status       = DOWN,
           Uptime       = -,
           Downtime     = -,
           Down to UP transitions = 0,
Backup Server :
           Status       = DOWN,
           Uptime       = -,
           Downtime     = -,
           Down to UP transitions = 0
```

Note. The status information for the primary and backup servers is displayed only if a RADIUS Health Check session is enabled for the server. See "Configuring RADIUS Health Check" on page 33-14 for more information.

Authorization Statistics

Authorization is a type of RADIUS authentication that is used for Authenticated Switch Access. When a user connects to the switch through console, Telnet, FTP, SSH, SNMP, HTTP, or HTTPS, access to manage the switch is authorized through the RADIUS server.

The following authorization information is displayed for each RADIUS server:

- Number of Access-Request
- Number of Access-Response
- Number of Access-Request timed out
- Last RTT of Access-Request and Access-Response
- Minimum RTT of Access-Request and Access-Response

- Average RTT of Access-Request and Access-Response
- Maximum RTT of Access-Request and Access-Response

Authentication Statistics

Authentication is used for network users and clients who use a RADIUS server for device authentication. A client may be a supplicant or non-supplicant device that is granted network access after successful authentication (802.1X authentication for supplicants; MAC authentication for non-supplicants) through the RADIUS server.

The following device authentication information is displayed for each RADIUS server and is cumulative for both 802.1X and MAC authentication:

- Number of Access-Request
- Number of Access-Response
  - Number of Access-Challenge
  - Number of Access-Accept
  - Number of Access-Reject
- Number of Access-Request timed out
- Last RTT of Access-Request and Access-Response
- Minimum RTT of Access-Request and Access-Response
- Average RTT of Access-Request and Access-Response
- Maximum RTT of Access-Request and Access-Response

Accounting Statistics

Accounting is used to collect global accounting statistics from defined RADIUS servers. Accounting statistics are collected for both ASA users and network access users.

The following information is displayed for each RADIUS server:

- Number of Accounting-Request
- Number of Accounting-Response
- Number of Accounting-Request timed out
- Last RTT of Accounting-Request and Access-Response
- Minimum RTT of Accounting-Request or Access-Response
- Average RTT of Accounting-Request or Access-Response
- Maximum RTT of Accounting-Request or Access-Response

BYOD Statistics

Bring Your Own Device (BYOD) statistics are related to asynchronous Change of Authorization (COA) and Disconnect Messages (DM) requests from the RADIUS server. The following BYOD statistics for each RADIUS server is displayed (the RTT is not calculated for COA or DM requests):

- Number of CoA Request (COA-Req)
- Number of CoA Acknowledgment (COA-ACK)
- Number of CoA Non-Acknowledgment (COA-NACK)
- Number of DM Request (DM-Req)
- Number of DM Acknowledgment Request (DM-ACK)
- Number of DM Non-Acknowledgment Request (DM-NACK)

Viewing the RADIUS Server Statistics

To view the authorization, authentication, accounting, and BYOD statistics for the configured RADIUS servers, use the **show aaa server statistics** command. For example:

```
-> show aaa server rad2 statistics
Statistics for rad2:
Authorization:
 Total No of Access-Request     : 2
 Total No of Access-Response    : 2
 Total No of Timedout Request   : 0
 Min RTT of Access Req/Res  usec: 938
 Avg RTT of Access Req/Res  usec: 1087
 Max RTT of Access Req/Res  usec: 1237
 Last RTT of Access Req/Res usec: 1237
Authentication:
 Total No of Access-Request     : 1
 Total No of Access-Response    : 1
 Total No of Access-Accept      : 1
 Total No of Access-Reject      : 0
 Total No of Access-Challenge   : 0
 Total No of Timedout Request   : 0
 Min RTT of Access Req/Res  usec: 1176
 Avg RTT of Access Req/Res  usec: 1176
 Max RTT of Access Req/Res  usec: 1176
 Last RTT of Access Req/Res usec: 1176
Accounting:
 Total No of Acct-Request       : 2
 Total No of Acct-Response      : 2
 Total No of Timedout Request   : 0
 Min RTT of Acct Req/Res    usec: 76657
 Avg RTT of Acct Req/Res    usec: 80182
 Max RTT of Acct Req/Res    usec: 83708
 Last RTT of Acct Req/Res   usec: 83708
BYOD:
 Total No of COA Request        : 0
 Total No of COA ACK Sent       : 0
 Total No of COA NACK Sent      : 0
 Total No of DM Request         : 0
 Total No of DM ACK Sent        : 0
 Total No of DM NACK Sent       : 0

Time of last statistics clear   : Thu Feb  1 18:09:06 2018
```

Clearing the RADIUS Server Statistics

Use the **aaa radius-server clear-statistics** command to clear RADIUS server statistics. For example:

```
-> clear radius-server rad2 clear-statistics
```

Setting UNP Profile Precedence

Use this command to set the precedence to filter ID or tunnel private group ID attributes for selection of UNP profile in the event of both these attributes being returned from the RADIUS server. By default, filter ID will be given precedence over the tunnel private group ID.

To set the UNP profile precedence, use **aaa radius unp-profile-precedence** command.

```
-> aaa radius unp-profile-precedence tunnel-private-group-id

-> aaa radius unp-profile-precedence filter-id
```

TACACS+ Server

Terminal Access Controller Access Control System (TACACS+) is a standard authentication and accounting protocol defined in RFC 1321 that employs TCP for reliable transport. A built-in TACACS+ client is available in the switch. A TACACS+ server allows access control for routers, network access servers, and other networked devices through one or more centralized servers. The protocol also allows separate authentication, authorization, and accounting services. By allowing arbitrary length and content authentication exchanges, it allows clients to use any authentication mechanism.

The TACACS+ client offers the ability to configure multiple TACACS+ servers. This can be done by the user. When the primary server fails, the client tries the subsequent servers. Multiple server configurations are applicable only for backup and not for server chaining.

In the TACACS+ protocol, the client queries the TACACS+ server by sending TACACS+ requests. The server responds with reply packets indicating the status of the request.

- **Authentication.** TACACS+ protocol provides authentication between the client and the server. It also ensures confidentiality because all the exchanges are encrypted. The protocol supports fixed passwords, one-time passwords, and challenge-response queries. Authentication is not a mandatory feature, and it can be enabled without authorization and accounting. During authentication if a user is not found on the primary TACACS+ server, the authentication fails. The client does not try to authenticate with the other servers in a multiple server configuration. If the authentication succeeds, then Authorization is performed.
- **Authorization.** Enabling authorization determines if the user has the authority to execute a specified command. TACACS+ authorization cannot be enabled independently. The TACACS+ authorization is enabled automatically when the TACACS+ authentication is enabled.
- **Accounting.** The process of recording what the user is attempting to do or what the user has done is Accounting. The TACACS+ accounting must be enabled on the switches for accounting to succeed. Accounting can be enabled irrespective of authentication and authorization. TACACS+ supports three types of accounting:

  Start Records—Indicate the service is about to begin.

  Stop Records—Indicates the services has just terminated.

  Update Records—Indicates the services are still being performed.

TACACS+ Client Limitations

The following limitation apply to this implementation of the TACACS+ client application:

- TACACS+ supports Authenticated Switch Access and cannot be used for user authentication.
- Authentication and Authorization are combined together and cannot be performed independently.
- On the fly, command authorization is not supported. Authorization is similar to the AOS partition management families.
- Only inbound ASCII logins are supported.
- A maximum of 50 simultaneous TACACS+ sessions can be supported when no other authentication mechanism is activated.
- Accounting of commands performed by the user on the remote TACACS+ process is not supported in the **boot.cfg** file at boot up time.

- TACAS authentication is not allowed on slave Console on a VC. Only local admin user on slave or secondary console can authenticate through TELNET access.
- System command supports only partition manager family based authorization.

Configuring the TACACS+ Client

Use the **aaa tacacs+-server** command to configure TACACS+ parameters on the switch.

TACACS+ server keywords	
key	**hash-salt**
host	**timeout**
prompt-key	**port**
salt	

When creating a new server, at least one host name or IP address (specified by the **host** keyword) is required as well as the shared secret (specified by the **key**, **hash-key**, or **prompt-key** keyword).

In this example, the server name is **tac1**, the host address is 10.10.5.2, the backup address is 10.10.5.5, and the shared secret is **otna**. Note that the shared secret must be configured exactly the same as on the server.

```
-> aaa tacacs+-server tac1 host 10.10.5.2 10.10.5.5 key otna
```

A **prompt-key** option is provided, which can be used to enter the secret key in an obscured format rather than as clear text. When this option is selected, press the Enter key. A password prompt appears prompting to enter the secret key. Secret key needs to be re-entered, and only if both the entries match, command is accepted. Key provided in this mode is not displayed on the CLI as text. For example:

```
-> aaa tacacs+-server tac1 prompt-key host 10.10.2.2
Enter Key:  *******
Confirm Key:   *******
```

Salt and hash-salt options are provided to add randomness for the encryption of key.

Use the **salt** option to add randomness to the encryption of key. The maximum length of the salt is 15 characters, and must be in clear text format. By default, system time (24-hour value format) will be taken as default salt value.

```
-> aaa tacacs+-server T1 host 10.10.10.3 key myorg salt salt@123
```

Note. To use a special character in the salt value, put the special character between double quotes ("").

Use the **hash-salt** option to enter the salt value in an encrypted format. The maximum length of the hash-salt is 64 characters.

```
-> aaa tacacs+-server tpub host 10.10.2.2 key otna hash-salt
c7f5eee2c0f9b33e72e3482673fb6059
```

To modify a TACACS+ server, enter the server name and the desired parameter to be modified.

```
-> aaa tacacs+-server tac1 key tnemelc
```

After entering the **aaa tacacs+-server** command to create or modify the server, you can use command prefix recognition. For example:

```
-> aaa tacacs+-server tac1 timeout 5
```

For information about server defaults, see "Server Defaults" on page 33-2.

To remove a TACACS+ server, use the **no** form of the command (delete only one server at a time):

```
-> no aaa tacacs+-server tac1
```

Managing Reauthentication for TACACS+ Sessions

The reauthentication request for a TACACS+ session can be managed by configuring the timeout interval for reauthentication. The timeout interval is entered in minutes. The timeout interval can be configured using the **session reauthentication** CLI. For example to configure a timeout interval of three minutes, enter:

```
-> session reauthentication 3
```

Note. The valid range is from 1 to 1200 minutes. By default, the timeout interval is set to 5 minutes.

The session timeout interval can be disabled by configuring the timeout value to 0. For example:

```
-> session reauthentication 0
```

The configuration status can be viewed using the **show session config** CLI.

LDAP Servers

Lightweight Directory Access Protocol (LDAP) is a standard directory server protocol. The LDAP client in the switch is based on several RFCs: 1798, 2247, 2251, 2252, 2253, 2254, 2255, and 2256. The protocol was developed as a way to use directory services over TCP/IP and to simplify the directory access protocol (DAP) defined as part of the Open Systems Interconnection (OSI) effort. Originally it was a front-end for X.500 DAP.

The protocol synchronizes and governs the communications between the LDAP client and the LDAP server. The protocol also dictates how its databases of information, which are normally stored in hierarchical form, are searched, from the root directory down to distinct entries.

In addition, LDAP has its own format that permits LDAP-enabled Web browsers to perform directory searches over TCP/IP.

Setting Up the LDAP Authentication Server

1 Install the directory server software on the server.

2 Copy the relevant schema LDIF files from the Alcatel-Lucent Enterprise software CD to the configuration directory on the server. (Each server type has a command line tool or a GUI tool for importing LDIF files.) Database LDIF files can also be copied and used as templates. The schema files and the database files are specific to the server type. The files available on the Alcatel-Lucent Enterprise software CD include the following:

aaa_schema.microsoft.ldif
aaa_schema.netscape.ldif
aaa_schema.novell.ldif
aaa_schema.openldap.schema
aaa_schema.sun.ldif

aaa_database.microsoft.ldif
aaa_database.netscape.ldif
aaa_database.novell.ldif
aaa_database.openldap.ldif
aaa_database.sun.ldif

3 After the server files have been imported, restart the server.

Note. Schema checking must be enabled on the server.

Information in the server files must match information configured on the switch through the **aaa ldap-server** command. For example, the port number configured on the server must be the same as the port number configured on the switch. See "Configuring the LDAP Authentication Client" on page 33-33 for information about using this command.

LDAP Server Details

LDAP servers must be configured with the properly defined LDAP schema and correct database suffix, including well-populated data. LDAP schema is extensible, permitting entry of user-defined schema as needed.

LDAP servers are also able to import and export directory databases using LDIF (LDAP Data Interchange Format).

LDIF File Structure

LDIF is used to transfer data to LDAP servers in order to build directories or modify LDAP databases. LDIF files specify multiple directory entries or changes to multiple entries, but not both. The file is in simple text format and can be created or modified in any text editor. In addition, LDIF files import and export binary data encoded according to the base 64 convention used with MIME (Multipurpose Internet Mail Extensions) to send various media file types, such as JPEG graphics, through electronic mail.

An LDIF file entry used to define an organizational unit would look like this:

dn: <distinguished name>
objectClass: top
objectClass: organizationalUnit
ou: <organizational unit name>
<list of optional attributes>

Below are definitions of some LDIF file entries:

entries	definition
dn: <distinguished name>	Defines the DN (required).
objectClass: top	Defines top object class (at least one is required). Object class defines the list of attributes required and allowed in directory server entries.
objectClass: organizationalUnit	Specifies that organizational unit must be part of the object class.
ou: <organizationalUnit name>	Defines the name of the organizational unit.
<list of attritbutes>	Defines the list of optional entry attributes.

Common Entries

The most common LDIF entries describe people in companies and organizations. The structure for such an entry might look like the following:

dn: <distinguished name>
objectClass: top
objectClass: person
objectClass: organizational Person
cn: <common name>
sn: <surname>
<list of optional attributes>

This is how the entry would appear with actual data in it.

dn: uid=yname, ou=people, o=yourcompany
objectClass: top
objectClass: person
objectClass: organizational Person
cn: your name
sn: last name
givenname: first name
uid: yname
ou: people
description:
<list of optional attributes>
. . .

Directory Entries

Directory entries are used to store data in directory servers. LDAP–enabled directory entries contain information about an object (person, place, or thing) in the form of a Distinguished Name (DN) that must be created in compliance with the LDAP protocol naming conventions.

Distinguished names are constructed from Relative Distinguished Names (RDNs), related entries that share no more than one attribute value with a DN. RDNs are the components of DNs, and DNs are string representations of entry names in directory servers.

Distinguished names typically consist of descriptive information about the entries they name, and frequently include the full names of individuals in a network, their email addresses, TCP/IP addresses, with related attributes such as a department name, used to further distinguish the DN. Entries include one or more object classes, and often a number of attributes that are defined by values.

Object classes define all required and optional attributes (a set of object classes is referred to as a "schema"). As a minimum, every entry must include the DN and one defined object class, like the name of an organization. Attributes required by a particular object class must also be defined. Some commonly used attributes that comprise a DN include the following:

Country (c), State or Province (st), Locality (l),
Organization (o), Organization Unit (ou),
and Common Name (cn)

Although each attribute would necessarily have its own values, the attribute syntax determines what kind of values are allowed for a particular attribute, e.g., (c=US), where country is the attribute and US is the value. Extra consideration for attribute language codes is required if entries are made in more than one language.

Entries are usually based on physical locations and established policies in a Directory Information Tree (DIT); the DN locates an entry in the hierarchy of the tree. Alias entries pointing to other entries can also be used to circumvent the hierarchy during searches for entries.

Once a directory is set up, DN attributes must thereafter be specified in the same order to keep the directory paths consistent. DN attributes are separated by commas as shown in this example:

cn=your name, ou=your function, o= your company, c=US

As there are other conventions used, please refer to the appropriate RFC specification for further details.

In addition to managing attributes in directory entries, LDAP makes the descriptive information stored in the entries accessible to other applications. The general structure of entries in a directory tree is shown in the following illustration. It also includes example entries at various branches in the tree.

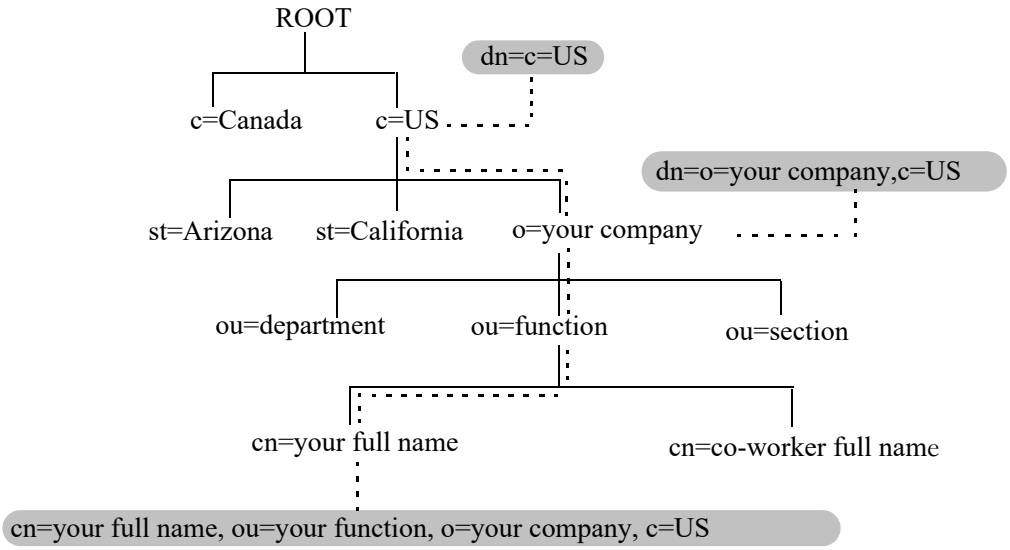


Figure 33-2 : Directory Information Tree

Directory Searches

DNs are always the starting point for searches unless indicated otherwise in the directory schema.

Searches involve the use of various criteria including scopes and filters which must be predefined, and utility routines, such as Sort. Searches must be limited in scope to specific durations and areas of the directory. Some other parameters used to control LDAP searches include the size of the search and whether to include attributes associated with name searches.

Base objects and scopes are specified in the searches, and indicate where to search in the directory. Filters are used to specify entries to select in a given scope. The filters are used to test the existence of object class attributes, and enable LDAP to emulate a “read” of entry listings during the searches. All search preferences are implemented by means of a filter in the search. Filtered searches are based on some component of the DN.

Retrieving Directory Search Results

Results of directory searches are individually delivered to the LDAP client. LDAP referrals to other servers are not returned to the LDAP client, only results or errors. If referrals are issued, the server is responsible for them, although the LDAP client retrieves results of asynchronous operations.

Directory Modifications

Modifications to directory entries contain changes to DN entry attribute values, and are submitted to the server by an LDAP client application. The LDAP-enabled directory server uses the DNs to find the entries to either add or modify their attribute values.

Attributes are automatically created for requests to add values if the attributes are not already contained in the entries.

All attributes are automatically deleted when requests to delete the last value of an attribute are submitted. Attributes can also be deleted by specifying delete value operations without attaching any values.

Modified attribute values are replaced with other given values by submitting replace requests to the server, which then translates and performs the requests.

Directory Compare and Sort

LDAP compares directory entries with given attribute values to find the information it needs. The Compare function in LDAP uses a DN as the identity of an entry, and searches the directory with the type and value of an attribute. Compare is similar to the Search function, but simpler.

LDAP also sorts entries by their types and attributes. For the Sort function, there are essentially two methods of sorting through directory entries. One is to sort by entries where the DN (Distinguished Name) is the sort key. The other is to sort by attributes with multiple values.

The LDAP URL

LDAP URLs are used to send search requests to directory servers over TCP/IP on the internet, using the protocol prefix: **ldap://.** (Searches over SSL would use the same prefix with an “s” at the end, i.e., **ldaps:**//.)

LDAP URLs are entered in the command line of any web browser, just as HTTP or FTP URLs are entered. When LDAP searches are initiated LDAP checks the validity of the LDAP URLs, parsing the various components contained within the URLs to process the searches. LDAP URLs can specify and implement complex or simple searches of a directory depending on what is submitted in the URLs. Searches performed directly with LDAP URLs are affected by the LDAP session parameters described above.

In the case of multiple directory servers, LDAP URLS are also used for referrals to other directory servers when a particular directory server does not contain any portion of requested IP address information. Search requests generated through LDAP URLs are not authenticated.

Searches are based on entries for attribute data pairs.

The syntax for TCP/IP LDAP URLs is as follows:

ldap://<hostname>:<port>/<base_dn>?attributes>?<scope>?<filter>

An example might be:

ldap://ldap.company name.xxx/o=company name%inc./,c=US>
(base search including all attributes/object classes in scope).

LDAP URLs use the percent symbol to represent commas in the DN. The following table shows the basic components of LDAP URLs.

components	description
<ldap>	Specifies TCP/IP connection for LDAP protocol. (The <ldaps> prefix specifies SSL connection for LDAP protocol.)
<hostname>	Host name of directory server or computer, or its IP address (in dotted decimal format).
<port>	TCP/IP port number for directory server. If using TCP/IP and default port number (389), port need not be specified in the URL. SSL port number for directory server (default is 636).

components	description
<base_dn>	DN of directory entry where search is initiated.
<attributes>	Attributes to be returned for entry search results. All attributes are returned if search attributes are not specified.
<scope>	Different results are retrieved depending on the scopes associated with entry searches. “base” search: retrieves information about distinguished name as specified in URL. This is a <base_dn> search. Base searches are assumed when the scope is not designated. “one” (one-level) search: retrieves information about entries one level under distinguished name (<base_dn> as specified in the URL, excluding the base entry. “sub” (subtree) search: retrieves information about entries from all levels under the distinguished name (<base_dn>) as specified in the URL, including the base entry.
<filter>	Search filters are applied to entries within specified search scopes. Default filter objectClass=* is used when filters are not designated. (Automatic search filtering not yet available.)

Password Policies and Directory Servers

Password policies applied to user accounts vary slightly from one directory server to another. Normally, only the password changing policies can be set by users through the directory server graphical user interface (GUI). Other policies accessible only to Network Administrators through the directory server GUI can include one or more of the following operational parameters.

- Log-in Restrictions
- Change Password
- Check Password Syntax
- Password Minimum Length
- Send Expiration Warnings
- Password History
- Account Lockout
- Reset Password Failure Count
- LDAP Error Messages (e.g., Invalid Username/Password, Server Data Error, etc.)

For instructions on installing LDAP-enabled directory servers, refer to the vendor-specific instructions.

Directory Server Schema for LDAP Authentication

Object classes and attributes need to be modified accordingly to include LDAP authentication in the network (object classes and attributes are used specifically here to map user account information contained in the directory servers).

- All LDAP-enabled directory servers require entry of an auxiliary objectClass:passwordObject for user password policy information.
- Another auxiliary objectClass: password policy is used by the directory server to apply the password policy for the entire server. There is only one entry of this object for the database server.

Note. Server schema extensions must be configured before the **aaa ldap-server** command is configured.

Vendor-Specific Attributes for LDAP Servers

The following are Vendor Specific Attributes (VSAs) for Authenticated Switch Access and/or Layer 2 Authentication:

attribute	description
bop-asa-func-priv-read-1	Read privileges for the user.
bop-asa-func-priv-read-2	Read privileges for the user.
bop-asa-func-priv-write-1	Write privileges for the user.
bop-asa-func-priv-write-2	Write privileges for the user.
bop-asa-allowed-access	Whether the user has access to configure the switch.
bop-asa-snmp-level-security	Whether the user can have SNMP access, and the type of SNMP protocol used.
bop-shakey	A key computed from the user password with the alp2key tool.
bop-md5key	A key computed from the user password with the alp2key tool.
allowedtime	The periods of time the user is allowed to log into the switch.
switchgroups	The VLAN ID and protocol (**IP_E2**, **IP_SNAP**, **IPX_E2**, **IPX_NOV**, **IPX_LLC**, **IPX_SNAP).**

Setting the SNMP Security Level

Use the table below to set the appropriate **bop-asa-snmp-level-security** attribute.

Level	LDAP snmp-level-security	Definition
no	1	No SNMP access allowed
no auth	2	SNMP access allowed without any SNMP authentication and encryption
sha	3	SHA authentication algorithm needed for authenticating SNMP
md5	4	MD5 authentication algorithm needed for authenticating SNMP
sha+des	5	SHA authentication algorithm and DES encryption needed for authentication SNMP
md5+des	6	MD5 authentication algorithm and DES encryption needed for authentication SNMP

Configuring Functional Privileges on the Server

Configuring the functional privileges attributes (**bop-asa-func-priv-read-1**, **bop-asa-func-priv-read-2**, **bop-asa-func-priv-write-1**, **bop-asa-func-priv-write-2**) requires using read and write bitmasks for command families on the switch.

1 To display the functional bitmasks of the desired command families, use the **show aaa priv hexa** command.

2 On the LDAP server, configure the functional privilege attributes with the bitmask values.

For more information about configuring users on the switch, see the Switch Security chapter of the *OmniSwitch AOS Release 8 Switch Management Guide*.

Configuring Authentication Key Attributes

The alp2key tool is provided on the Alcatel-Lucent Enterprise software CD for computing SNMP authentication keys.The alp2key application is supplied in two versions, one for Unix (Solaris 2.5.1 or higher) and one for Windows (NT 4.0 and higher).

To configure the bop-shakey or bop-md5key attributes on the server:

1 Use the alp2key application to calculate the authentication key from the password of the user. The switch automatically computes the authentication key, but for security reasons the key is never displayed in the CLI.

2 Cut and paste the key to the relevant attribute on the server.

An example using the alp2key tool to compute the SHA and MD5 keys for **mypassword**:

```
ors40595{}128: alp2key mypassword

bop-shakey: 0xb1112e3472ae836ec2b4d3f453023b9853d9d07c
bop-md5key: 0xeb3ad6ba929441a0ff64083d021c07f1
ors40595{}129:
```

Note. The bop-shakey and bop-md5key values must be recomputed and copied to the server any time a user password is changed.

LDAP Accounting Attributes

Logging and accounting features include Account Start, Stop and Fail Times, and Dynamic Log. Typically, the Login and Logout logs can be accessed from the directory server software. Additional third-party software is required to retrieve and reset the log information to the directory servers for billing purposes.

The following sections describe accounting server attributes.

AccountStartTime

User account start times are tracked in the AccountStartTime attribute of the directory entry of the user that keeps the time stamp and accounting information of user log-ins. The following fields (separated by carriage returns "|") are contained in the Login log. Some fields are only used for Layer 2 Authentication.

Fields Included For Any Type of Authentication

- User account ID or username client entered to log-in: variable length digits.
- Time Stamp (YYYYMMDDHHMMSS (YYYY:year, MM:month, DD:day, HH:hour, MM:minute, SS:second)
- Switch serial number: Alcatel-Lucent.BOP.<*switch name*>.<*MAC address*>
- Client IP address: variable length digits.

Fields Included for Layer 2 Authentication Only

- Client MAC address: xx:xx:xx:xx:xx:xx:xx (alphanumeric).
- Switch VLAN number client joins in multiple authority mode (0=single authority; 2=multiple authority); variable-length digits.
- Switch slot number to which client connects: nn
- Switch port number to which client connects: nn
- Switch virtual interface to which client connects: nn

AccountStopTime

User account stop times are tracked in the AccountStopTime attribute that keeps the time stamp and accounting information of successful user log-outs. The same fields as above (separated by carriage returns "|") are contained in the Logout log. A different carriage return such as the # sign can be used in some situations. Additionally, these fields are included but apply only to the Logout log:

Fields For Any Type of Authentication

- Log-out reason code, for example LOGOFF(18) or DISCONNECTED BY ADMIN(19)
- User account ID or username client entered to log-in: variable length digits.

Fields For Layer 2 Authentication Only

- Number of bytes received on the port during the client session from log-in to log-out: variable length digits.
- Number of bytes sent on the port during the client session from log-in to log-out: variable length digits.

- Number of frames received on the port during the client session from log-in to log-out: variable length digits.
- Number of frames sent on the port during the client session from log-in to log-out: variable length digits.

AccountFailTime

The AccountFailTime attribute log records the time stamp and accounting information of unsuccessful user log-ins. The same fields in the Login Log—which are also part of the Logout log (separated by carriage returns "|")—are contained in the Login Fail log. A different carriage return such as the # sign can be used in some situations. Additionally, these fields are included but apply only to the Login Fail log.

- User account ID or username client entered to log-in: variable length digits.
- Log-in fail error code: nn. For error code descriptions refer to the vendor-specific listing for the specific directory server in use.
- Log-out reason code, for example PASSWORD EXPIRED(7) or AUTHENTICATION FAILURE(21).

Dynamic Logging

Dynamic logging can be performed by an LDAP-enabled directory server if an LDAP server is configured **first** in the list of authentication servers configured through the **aaa accounting session** command. Any other servers configured are used for accounting (storing history records) only. For example:

```
-> aaa accounting session ldap2 rad1 rad2
```

In this example, server **ldap2** is used for dynamic logging, and servers **rad1** and **rad2** is used for accounting.

If you specify a RADIUS server first, all of the servers specified is used for recording history records (not logging). For example:

```
-> aaa accounting session rad1 ldap2
```

In this example, both the **rad1** and **ldap2** servers is used for history only. Dynamic logging does not take place on the LDAP server.

Dynamic entries are stored in the LDAP-enabled directory server database from the time the user successfully logs in until the user logs out. The entries are removed when the user logs out.

- Entries are associated with the switch the user is logged into.
- Each dynamic entry contains information about the user connection. The related attribute in the server is bop-loggedusers.

A specific object class called **alcatelBopSwitchLogging** contains three attributes as follows:

Attribute	Description
bop-basemac	MAC range, which uniquely identifies the switch.
bop-switchname	Host name of the switch.
bop-loggedusers	Current activity records for every user logged onto the switch identified by bop-basemac.

Each switch that is connected to the LDAP-enabled directory server has a DN starting with bop-basemac-*xxxxx*, ou=bop-logging. If the organizational unit ou=bop.logging exists somewhere in the tree under searchbase, logging records are written on the server. See the documentation of the server manufacturer for more information about setting up the server.

The bop-loggedusers attribute is a formatted string with the following syntax:

loggingMode : accessType ipAddress port macAddress vlanList userName

The fields are defined here:

Field	Possible Values
loggingMode	**ASA** *x*—for an authenticated user session, where *x* is the number of the session **AVLAN**—for Authenticated VLAN session in single authority mode **AVLAN** *y*—for Authenticated VLAN session in multiple authority mode, where *y* is relevant VLAN
accessType	Any one of the following: **CONSOLE**, **MODEM**, **TELNET**, **HTTP**, **FTP**, **XCAP**
ipAddress	The string **IP** followed by the IP address of the user.
port	(For Authenticated VLAN users only.) The string **PORT** followed by the slot/port number.
macAddress	(For Authenticated VLAN users only.) The string **MAC** followed by the MAC address of the user.
vlanList	(For Authenticated VLAN users only.) The string **VLAN** followed by the list of VLANs the user is authorized (for single-mode authority).
userName	The login name of the user.

For example:

```
"ASA      0   :  CONSOLE IP 65.97.233.108   Jones"
```

Configuring the LDAP Authentication Client

Use the **aaa tacacs+-server** command to configure LDAP authentication parameters on the switch. The server name, host name or IP address, distinguished name, password, and the search base name are required for setting up the server. Optionally, a backup host name or IP address can be configured, as well as the number of retransmit tries, the timeout for authentication requests, and whether or not a secure Socket Layer (SSL) is enabled between the switch and the server.

Note. The server must be configured with the appropriate schema before the **aaa ldap-server** command is configured.

The keywords for the **aaa ldap-server** command are listed here:

Required for creating:	optional:
host	**type**
dn	**retransmit**
password	**timeout**
base	**port**
	ssl

Creating an LDAP Authentication Server

When creating a server, at least one-host name or IP address (specified by the **host** keyword), distinguished name, search base recognized by the LDAP-enabled directory server is required as well as the super user password (specified by the **password**, **hash-password**, or **prompt-password** keyword).

An example of creating an LDAP server:

```
-> aaa ldap-server ldap2 host 10.10.3.4 dn cn=manager password tpub base c=us
```

In this example, the switch can communicate with an LDAP server (called **ldap2**) that has an IP address of 10.10.3.4, a domain name of cn=manager, a password of tpub, and a searchbase of c=us. These parameters must match the same parameters configured on the server itself.

Note. The distinguished name must be different from the searchbase name.

An option **prompt-password** is provided, which can be used to enter the super-user password in a obscured format rather than as clear text. When this option is selected, press the Enter key. A password prompt appears prompting to enter the super-user password. Password needs to be re-entered, and only if both the passwords match, command is accepted. Password provided in this mode is not displayed on the CLI as text. For example:

```
-> aaa ldap-server topanga5 host 10.10.3.4 dn cn=manager prompt-password base c=us
retransmit 4
Enter Password:  *******
Confirm Password:   *******
```

Salt and hash-salt option are provided to add randomness for the encryption of key.

Use the **salt** option to add randomness to the encryption of key. The maximum length of the salt is 15 characters, and must be in clear text format. By default, system time (24-hour value format) will be taken as default salt value. The user configured or default salt along with the server name will be combined with 'key' and encrypted as a whole, the output of which will be displayed under ‘hash-key'.

```
-> aaa ldap-server topanga5 host 10.10.3.4 dn cn=manager hash-password
c7f5eee2c0f9b33e72e3482673fb6059 salt random base c=us
```

Note. To use a special character in the salt value, put the special character between double quotes ("").

Use the **hash-salt** option to enter the salt value in an encrypted format. The maximum length of the hash-salt is 64 characters.

```
-> aaa ldap-server topanga5 host 10.10.3.4 dn cn=manager hash-password
c7f5eee2c0f9b33e72e3482673fb6059 hash-salt c7f5eee2c0f9b33e72e3482673fb6059 base
c=us
```

Modifying an LDAP Authentication Server

To modify an LDAP authentication server, use the **aaa ldap-server** command with the server name; or, if you have just entered the **aaa ldap-server** command to create or modify the server, you can use command prefix recognition. For example:

```
-> aaa ldap-server ldap2 password my_pass

-> timeout 4
```

In this example, an existing LDAP server is modified with a different password, and then the timeout is modified on a separate line. These two command lines are equivalent to:

```
-> aaa ldap-server ldap2 password my_pass timeout 4
```

Setting Up SSL for an LDAP Authentication Server

A Secure Socket Layer (SSL) can be set up on the server for additional security. When SSL is enabled, the server identity is authenticated. The authentication requires a certificate from a Certification Authority (CA). If the CA providing the certificate is well-known, the certificate is automatically extracted from the **Kbase.img** file on the switch (**certs.pem**). If the CA is not well-known, the CA certificate must be transferred to the switch through FTP to the /flash/certified or /flash/working directory and must be named **optcerts.pem**. The switch merges either or both of these files into a file called **ldapcerts.pem**.

To set up SSL on the server, specify **ssl** with the **aaa ldap-server** command:

```
-> aaa ldap-server ldap2 ssl
```

The switch automatically sets the port number to 636 when SSL is enabled. The 636 port number is typically used on LDAP servers for SSL. The port number on the switch must match the port number configured on the server. If the port number on the server is different from the default, use the **aaa ldap-server** command with the **port** keyword to configure the port number. For example, if the server port number is 635, enter the following:

```
-> aaa ldap-server ldap2 port 635
```

The switch can now communicate with the server on port 635.

To remove SSL from the server, use **no** with the **ssl** keyword. For example:

```
-> aaa ldap-server ldap2 no ssl
```

SSL is now disabled for the server.

Removing an LDAP Authentication Server

To delete an LDAP server from the switch configuration, use the **no** form of the command with the relevant server name.

```
-> no aaa ldap-server topanga5
```

The topanga5 server is removed from the configuration.

Configuring OpenSSL Ciphers

Many applications use OpenSSL to communicate to external network elements (over TLS). OpenSSL allows the application to select their own cipher suites (a list of cryptography algorithms which will be used for the connection establishment, key exchange and data encryption).

Most of the applications using the OpenSSL do not share common cipher suites, which make it difficult for the network administrator to know which cipher suite is used by which application.

Open SSL cipher security level configuration allows to configure common SSL cipher suites for RADIUS, LDAP, Captive Portal, Syslog and SNMP applications which use OpenSSL to communicate over TLS.

OpenSSL cipher security level configuration provides four security levels for the network administrator to choose from. Each level specifies the strength of the cipher and indicates the minimum level of ciphers that are supported. The following security levels can be configured:

- **All**: Includes all the cipher suites, including NULL-SHA.
- **Low**: Includes all cipher suites, except NULL-SHA.
- **Medium**: Includes all ciphers suites except NULL-SHA, DES-CBC-SHA, and RC4-MD5.
- **High**: Includes only AES-256 with SHA-2 ciphers (Applicable only for TLSv1.2).

By default, the cipher security level is set to medium in default switch operation mode and high in common criteria mode.

Apart from the predefined cipher security level, the administrator can also define custom cipher suites as per requirement using the custom configuration.

Configuring Cipher Security Level

The cipher security level can be configured using the **ssl cipher** command. For example, to set the security level to high, enter:

```
-> ssl cipher level high
```

After running the command, the switch must be rebooted for the security level to be applied.

Configuring Custom Cipher Suite

The custom cipher suite can be configured using the **ssl cipher** command with the custom option. For example, to set the cipher suite AECDH-AES256-SHA enter:

```
-> ssl cipher custom AECDH-AES256-SHA
```

After running the command, the switch must be rebooted for the custom cipher suite to be applied.

The Custom ciphers cannot be more than 255 characters. Hence, the cipher files can be used to configure ciphers more than 255 characters. The cipher file can be created by copying the required ciphers in the notepad and saving it as a cipher file with "**.cipher**" as the file extension. The cipher file must be copied to the flash directory of the switch before using this command.

In chassis based model, the cipher file needs to be copied in both the primary and secondary unit flash directory. In VC based models, the cipher file must be copied to all the flash directory of all the modules.

For example, to set the cipher file abc.cipher enter:

```
-> ssl cipher custom file /flash/abc.cipher
```

Note. The custom cipher suite must be configured based on the supported ciphers. To view the supported ciphers, use the **show ssl ciphers all** command.

Viewing the Cipher Suite Configuration

To view the cipher suite configuration, use the **show ssl ciphers config** command. For example:

```
-> show ssl ciphers config

SSL Cipher Global Configuration:
SSL Cipher Level = medium
SSL Cipher Suite = ALL:eNULL:!NULL-SHA:!DES-CBC-SHA:!RC4-MD5
```

For more information about the CLI commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Public Key Infrastructure (PKI)

Applications using OpenSSL can select the public key to communicate with external servers when servers require to verify client certificate. Likewise, clients can also validate the server certificate. This prevents the spoofing attacks.

The following three public key security modes can be configured for TLS client to communicate with external servers:

- **No Validation**: This is the default mode, in this mode the client applications do not provide certificate and not validate server certificate.
- **Server Certificate Validation**: In this mode, the client application is required to provide clients certificate but the client will validate the server certificate using the pre-installed CA certificate.
- **Mutual Authentication**: In this mode, the client application must load their certificates and key files and provide clients certificate to server.

The applications can also limit the TLS version it uses.

The PKI feature allows to select common certificate and public key security mode and configure the TLS version for the applications (RADIUS, LDAP, Captive Portal, Syslog and SNMP) using OpenSSL.

Configuring Server Certificate Validation

The server certificate validation can be enabled using the **ssl pki client validate-certificate admin-state** CLI command. For example:

```
-> ssl pki client validate-certificate admin-state enable
```

When the feature is enabled or disabled the switch must be rebooted for the changes to be applied.

When the server validation is enabled, the TLS client (LDAP, RADIIUS, SYSLOG) applications validate server certificate.

The server certificate is validated based on:

- TLS mutual authentication using X.509 certificates.
- The presented identifier must match the reference identifier as per RFC 6125 Section 6.
- X.509 certificate validation using OCSP and CRL.

Configuring Mutual Authentication

Mutual Authentication can be configured for client and server. To configure the mutual authentication for client, use the **ssl pki client mutual -authentication admin-state** CLI command. For example:

```
-> ssl pki client mutual-authentication admin-state enable
```

When the feature is enabled or disabled the switch must be rebooted for the changes to be applied.

When the mutual authentication is enabled for the client, the TLS client applications will load the **myCliCert.pem** and **myCliPrivate.key** files in **/flash/switch/cert.d** and provide the certificate to server while establishing the TLS connection.

If the server certificate is not validated, then the TLS client connection is terminated.

To configure the mutual authentication for the server, use the **ssl pki server mutual-authentication admin-state** CLI command. For example:

```
-> ssl pki server mutual-authentication admin-state enable
```

When the feature is enabled or disabled the switch must be rebooted for the changes to be applied.

When the mutual authentication is enabled for the server, the TLS server (SNMP) application must require clients to provide their certificate to server while establishing TLS connection.

The server certificate is validated based on:

- TLS mutual authentication using X.509 certificates.
- The presented identifier must match the reference identifier as per RFC 6125 Section 6.
- X.509 certificate validation using OCSP and CRL.

If the client certificate is not validated, then the TLS server connection is terminated.

Configuring TLS Version

The TLS version can be configured for server and client applications. When the version is configured the TLS client and server will deny all SSL and TLS versions lower than the configured version.

The TLS version can be configured using the **ssl pki tls version** CLI command. For example, to set the TLS version to 1.1 enter:

```
-> ssl pki tls version 1.1
```

The switch must be rebooted for the changes to be applied.

The TLS version can be set to 1.0 or 1.1 or 1.2.

The command is applicable only for LDAP, RADIUS, SYSLOG and SNMP applications.

Viewing the Public Key Infrastructure (PKI) Configuration

The PKI configuration can be viewed using the show **ssl pki config** CLI command. For example:

```
-> show ssl pki config

SSL PKI Global Configuration:
Client Validate Certificate = enabled
Client Mutual Authentication = enabled
Server Mutual Authentication = enabled
TLS version = 1.1
```

For more information about the CLI commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Verifying the Authentication Server Configuration

To display information about authentication servers, use the following command:

show aaa server	Displays information about a particular AAA server or AAA servers.

An example of the output for this command is given in "Quick Steps For Configuring Authentication Servers" on page 33-4. For more information about the output of this command, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Kerberos Snooping Overview

Kerberos is a secure method for authenticating a request for a service in a computer network. The purpose of Kerberos is to perform authentication between a client and a server.

Authentication is a mechanism whereby systems securely identify their users. Authentication provides a network security mechanism that is designed to check the identity of the client. Kerberos uses shared key cryptography in which both the user and the server have access to the same key, or password, used to positively identify the user.

The Kerberos protocol is designed to provide reliable authentication over open and insecure networks where communication between the hosts belonging to it may be intercepted. It is a robust security protocol used to establish the identity of users and systems accessing services across the network, to protect network protocols from tampering (integrity protection), and often to encrypt the data sent across the protocol (privacy protection).

It is based on the concept of symmetric encryption keys, which means that the same key is used to encrypt and decrypt a message. This is also referred to as a shared private key. It is a client-server based secret-key network authentication method that uses a trusted Kerberos server to verify secure access to both services and users. In Kerberos, this trusted server is called the key distribution center (KDC). The KDC issues tickets to validate users and services. The password of the user is never stored in any form on the client machine. The password is immediately discarded after being used.

Kerberos provides authentication only. It does not support user authorization.

Importance of Kerberos Authentication

In the password based authentication, passwords sent across the network can be intercepted and subsequently used by eavesdropper to impersonate the user. In addition to the security concern, password based authentication is inconvenient as users do not want to enter a password each time they access a network service.

With Kerberos authentication, user password is never sent across the network, encrypted or in plain text. Secret keys are only passed across the network in encrypted form. A user has to only authenticate to the Kerberos system once (using the principal and password). It provides single-sign-on, which lets a user log in to a system and access multiple systems or applications for a longer period without the need to enter the user name and password multiple times.

Kerberos Snooping Authentication

Kerberos snooping snoops the user information and identifies if a system has successfully logged on to a domain. Kerberos authentication is handled by external Kerberos server (KDC). Kerberos agent is placed between the client and the Kerberos server.

Kerberos agent maintains the database of the clients, that is, the client information (client name, source MAC address, IP address, and domain name), authenticated state, port number on which the client is attached, QoS policy-list that needs to be applied after authentication process is over.

The following example illustrates the Kerberos snooping scenario.

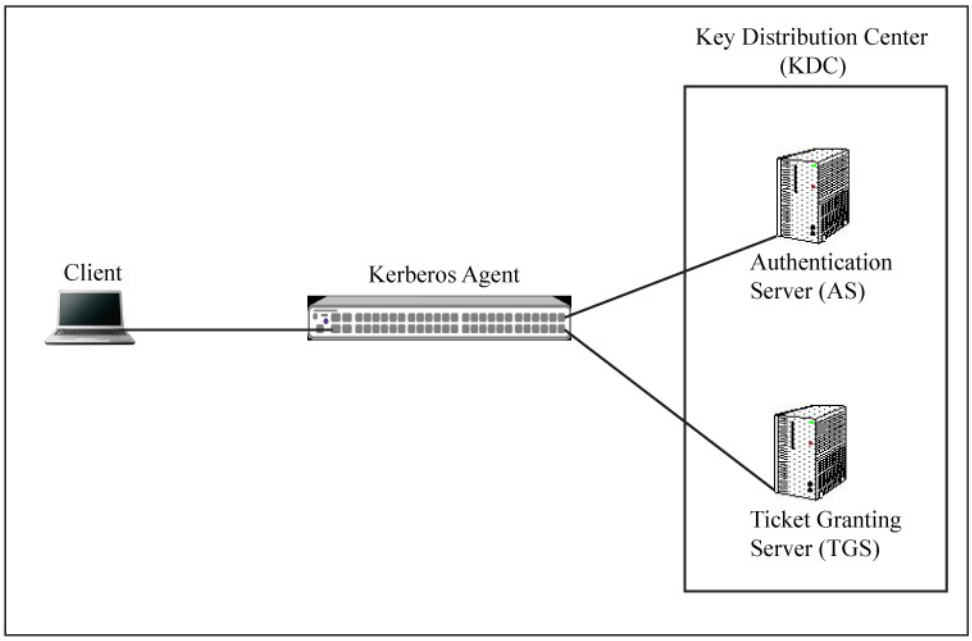


Figure 33-3 : Kerberos Snooping Authentication

Upon receiving the Kerberos Request Protocol Data Unit (PDU), Kerberos agent relays and snoops the authentication frames coming from the client and creates or updates the user entry. On reception of the request packet, KDC replies to the client by sending a response packet. Kerberos agent relays and snoops the reply packet coming from KDC and updates the authentication state of the client (authentication pass or fail). Once the client is authenticated successfully, and the user domain is classified under some QoS policy, then that QoS policy is applied.

Configuring Kerberos Snooping

This section describes how to configure Kerberos snooping using the CLI commands.

Enabling Kerberos Snooping

Kerberos snooping is supported only on UNP ports and the user should have learned into a UNP profile (MAC/Classification/default-profile) which has Kerberos authentication enabled. The UNP profile should be mapped to a VLAN.

To enable Kerberos snooping on a UNP profile, use the **unp profile kerberos-authentication** command at the CLI prompt as shown:

```
-> unp profile p1 kerberos-authentication
```

To disable Kerberos snooping on a UNP profile, use the **no** form of this command as shown:

```
-> no unp profile p1 kerberos-authentication
```

Note. Kerberos and BYOD will not be supported on the same UNP profile.

Configuring Kerberos Server

Kerberos server or Key Distribution Centre (KDC) runs on a network host that allocates the Kerberos credentials to different users or network services. These credentials are created by using information that is stored in the KDC database.

One Kerberos server and one Kerberos enabled port must be configured on the switch for Kerberos snooping to function. A maximum of two Kerberos servers can be configured on a switch.

To configure IP address of the Kerberos server and UDP/TCP port number, use the **kerberos ip-address** command at the CLI prompt as shown:

```
-> kerberos ip-address 172.21.160.102 port 2001
```

Note. Server IP address cannot be configured as 0.0.0.0, and the octet value in the IP address cannot be greater than 255 (for example, 1.256.2.3).

Use the **port** keyword to configure both UDP and TCP protocol port number.

Use the **no** form of this command to delete the Kerberos server IP address. Only one server can be deleted at a time.

```
-> no kerberos ip-address 172.21.160.102
```

Configuring Kerberos Inactivity Timer

Whenever a Kerberos user becomes inactive, inactivity timer is started for that user. If Kerberos user becomes active before the inactivity timer expiry, then the timer stops. Else, on timer expiry, user entry is removed from the Kerberos user database and inactivity timer trap is raised.

All inactive Kerberos user entries are visited every five minutes and the left-time value would be decremented by the elapsed time. If the total remaining time is equal to zero or less than zero, then

Kerberos user entry would be deleted from the system and corresponding QoS policy would be removed. In this approach, timer expiry can vary from five minutes to ten minutes from the expected result.

To configure global inactivity timer on the switch for Kerberos users, use the **kerberos inactivity-timer** command at the CLI prompt as shown:

```
-> kerberos inactivity-timer 30
```

By default, inactivity timer is set to 300 minutes.

Configuring Kerberos Server Timeout

All the users trying to get authenticated from a specific server has the same value for reply-timeout timer. Whenever a Kerberos request packet is sent to the server, the server reply time-out starts. If the timer expires before receiving the reply from the server, the user authentication is marked as server-time-out.

To configure global server reply time-out timer value on the switch for Kerberos users, use the **kerberos server-timeout** command at the CLI prompt as shown:

```
-> kerberos server-timeout 20
```

By default, reply-timeout is 2 seconds.

Configuring a Global Policy List for Kerberos Users

QoS policy list must be created prior to associating the policy list for Kerberos users. Per-user Kerberos policy list configuration is not supported.

To configure a global classification QoS policy list on the switch for Kerberos users, use the **kerberos authentication-pass policy-list-name** command at the CLI prompt as shown:

```
-> kerberos authentication-pass policy-list-name pl1
```

Use the **no** form of this command to the remove the global classification QoS policy list from the switch.

```
-> no kerberos authentication-pass policy-list-name
```

Consider the following additional information about the policy list association with Kerberos users:

- If a domain level policy list is configured on the switch and any user belonging to that domain gets authenticated from the Kerberos server, then the domain level policy list is applied to the users instead of the global policy list.
- If a user gets authenticated from the Kerberos server and the domain policy list is not configured on the switch for the authenticated user domain, then the global policy list is applied to the users if the globally policy list is configured on the switch.
- If a user gets authenticated from the Kerberos server and neither the domain policy list (for that user domain) nor the global policy list is configured, then the user traffic is classified on the basis of already applied non-supplicant authentication classification.

Configuring a Domain Policy List for Kerberos Users

To configure a domain classification policy for Kerberos users, use the **kerberos authentication-pass domain** command at the CLI prompt as shown:

```
-> kerberos authentication-pass domain EXAMPLE.COM policy-list-name p1
```

Use the **no** form of this command to remove the domain classification policy for Kerberos users.

```
-> aaa kerberos authentication-pass domain EXAMPLE.COM
```

Verifying Kerberos Snooping Configuration

A summary of the commands used for verifying the Kerberos Snooping configuration is given here:

show kerberos configuration	Displays Kerberos global configuration.
show kerberos users	Displays the learned Kerberos users information.
show kerberos statistics	Displays the global Kerberos statistics.

To clear global and port level Kerberos statistics, use the **clear kerberos statistics** command.

Note. The **show configuration snapshot da-unp** command displays the Kerberos configuration as shown below:

```
-> show configuration snapshot da-unp
! DA-UNP:
kerberos inactivity-timer 60
kerberos server-timeout 3
kerberos authentication-pass policy-list-name kpol
kerberos ip-address 33.33.33.10 port 88
kerberos authentication-pass domain KERBEROSTEST.COM policy-list-name
kpol_dom
```

For more information about the output details that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

34 Configuring Port Mapping

Port Mapping is a security feature that controls communication between peer users. Each session comprises of a session ID, a set of user ports, and/or a set of network ports. The user ports within a session cannot communicate with each other and can only communicate through network ports. In a port mapping session with user port set A and network port set B, the ports in set A can only communicate with the ports in set B. If set B is empty, the ports in set A can communicate with rest of the ports in the system.

A port mapping session can be configured in the unidirectional or bidirectional mode. In the unidirectional mode, the network ports can communicate with each other within the session. In the bidirectional mode, the network ports cannot communicate with each other. Network ports of a unidirectional port mapping session can be shared with other unidirectional sessions, but cannot be shared with any sessions configured in the bidirectional mode. Network ports of different sessions can communicate with each other.

In This Chapter

This chapter describes the port mapping security feature and explains how to configure the same through the Command Line Interface (CLI).

Configuration procedures described in this chapter include:

- Creating/Deleting a Port Mapping Session—see "Creating a Port Mapping Session" on page 34-4 or "Deleting a Port Mapping Session" on page 34-4.
- Enabling/Disabling a Port Mapping Session—see "Enabling a Port Mapping Session" on page 34-5 or "Disabling a Port Mapping Session" on page 34-5.
- Configuring a Port Mapping Direction—see "Configuring Unidirectional Port Mapping" on page 34-5 and "Restoring Bidirectional Port Mapping" on page 34-5.
- Configuring an example Port Mapping Session—see "Sample Port Mapping Configuration" on page 34-6.
- Verifying a Port Mapping Session—see "Verifying the Port Mapping Configuration" on page 34-7.

Port Mapping Defaults

The following table shows port mapping default values.

Parameter Description	CLI Command	Default Value/Comments
Mapping Session Creation	**port-mapping user-port network-port**	No mapping sessions
Mapping Status configuration	**port-mapping**	Disabled
Port Mapping Direction	**port-mapping unidirectional bidirectional**	Bidirectional
Port Mapping Unknown Unicast Flooding	**port-mapping unknown-unicast-flooding**	Enabled

Quick Steps for Configuring Port Mapping

Follow the steps below for a quick tutorial on configuring port mapping sessions. Additional information on how to configure each command is given in the subsections that follow.

1 Create a port mapping session with the user ports, network ports, or both user ports and network ports with the **port-mapping user-port network-port** command. For example:

```
-> port-mapping 8 user-port 1/2 network-port 1/3
```

2 Enable the port mapping session with the **port-mapping** command. For example:

```
-> port-mapping 8 enable
```

Note. You can verify the configuration of the port mapping session by entering **show port-mapping** followed by the session ID.

```
-> show port-mapping 8
SessionID       USR-PORT        NETWORK-PORT
-----------+----------------+------------------
     8             1/2             1/3
```

You can also verify the status of a port mapping session by using the **show port-mapping status** command.

Creating/Deleting a Port Mapping Session

Before port mapping can be used, it is necessary to create a port mapping session. The following subsections describe how to create and delete a port mapping session with the **port-mapping user-port network-port** and **port-mapping** command, respectively.

Creating a Port Mapping Session

To create a port mapping session either with the user ports, network ports, or both the user ports and network ports, use the **port-mapping user-port network-port** command. For example, to create a port mapping session 8 with a user port on slot 1 port 2 to port 5 and a network port on slot 2 port 3, enter:

```
-> port-mapping 8 user-port 1/2-5 network-port 2/3
```

You can create a port mapping session with link aggregate network ports. For example, to create a port mapping session 3 with network ports of link aggregation group 7 to 9, enter:

```
-> port-mapping 3 network-port linkagg 7
-> port-mapping 3 network-port linkagg 8
-> port-mapping 3 network-port linkagg 9
```

You can specify all the ports of a slot to be assigned to a mapping session. For example, to create a port mapping session 3 with all the ports of slot 1 as network ports, enter:

```
-> port-mapping 3 network-port slot 1
```

You can specify a range of ports to be assigned to a mapping session. For example, to create a port mapping session 4 with ports 5 through 8 on slot 2 as user ports, enter:

```
-> port-mapping 4 user-port 2/5-8
```

Deleting a User/Network Port of a Session

To delete a user/network port of a port mapping session, use the **no** form of the **port-mapping user-port network-port** command. For example, to delete a user port on slot 1 port 3 of a mapping session 8, enter:

```
-> no port-mapping 8 user-port 1/3
```

Similarly, to delete the network ports of link aggregation group 7 of a mapping session 4, enter:

```
-> no port-mapping 4 network-port linkagg 7
```

Deleting a Port Mapping Session

To delete a previously created mapping session, use the **no** form of the **port-mapping** command. For example, to delete the port mapping session 6, enter:

```
-> no port-mapping 6
```

Enabling/Disabling a Port Mapping Session

By default, the port mapping session is disabled. The following subsections describe how to enable and disable the port mapping session with the **port-mapping** command.

Enabling a Port Mapping Session

To enable a port mapping session, enter **port-mapping** followed by the session ID and **enable**. For example, to enable the port mapping session 5, enter:

```
-> port-mapping 5 enable
```

Disabling a Port Mapping Session

To disable a port mapping session, enter **port-mapping** followed by the session ID and **disable**. For example, to disable the port mapping session 5, enter:

```
-> port-mapping 5 disable
```

Disabling the Flooding of Unknown Unicast Traffic

By default, unknown unicast traffic is flooded to the user ports of a port mapping session from all the switch ports. To disable this flooding and to receive traffic from only the network ports, enter:

```
-> port-mapping 5 unknown-unicast-flooding disable
```

Configuring a Port Mapping Direction

By default, port mapping sessions are bidirectional. The following subsections describe how to configure and restore the directional mode of a port mapping session with the **port-mapping unidirectional bidirectional** command.

Configuring Unidirectional Port Mapping

To configure a unidirectional port mapping session, enter **port-mapping** followed by the session ID and **unidirectional** keyword. For example, to configure the direction of a port mapping session 6 as unidirectional, enter:

```
-> port-mapping 6 unidirectional
```

Restoring Bidirectional Port Mapping

To restore the direction of a port mapping session to its default (bidirectional), enter **port-mapping** followed by the session ID and **bidirectional** keyword. For example, to restore the direction (bidirectional) of the port mapping session 5, enter:

```
-> port-mapping 5 bidirectional
```

Note. To change the direction of an active session with network ports, delete the network ports of the session, change the direction, and recreate the network ports.

Sample Port Mapping Configuration

This section provides an example port mapping network configuration. In addition, a tutorial is also included that provides steps on how to configure the example port mapping session using the Command Line Interface (CLI).

Example Port Mapping Overview

The following diagram shows a four-switch network configuration with active port mapping sessions. In the network diagram, the Switch A is configured as follows:

- Port mapping session 1 is created with user ports 2/1, 2/2 and network ports 1/1, 1/2 and is configured in the unidirectional mode.
- Port mapping session 2 is created with user ports 3/1, 3/2, and 3/3 and network port 1/3.

The Switch D is configured by creating a port mapping session 1 with user ports 2/1, 2/2 and network ports 1/1.

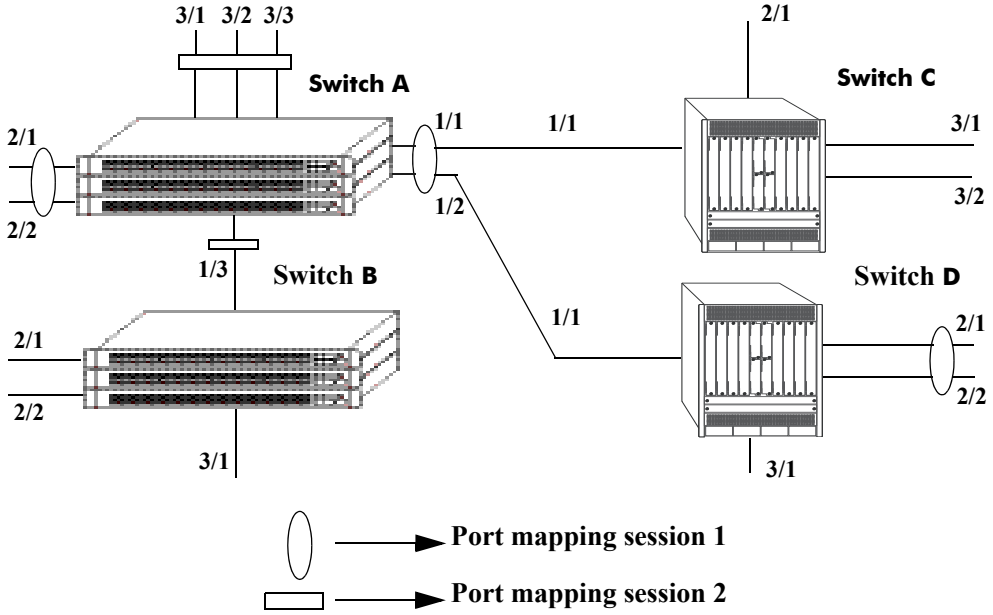


Figure 34-1 : Example Port Mapping Topology

In the above example topology:

- Ports 2/1 and 2/2 on Switch A do not interact with each other and do not interact with the ports on Switch B.
- Ports 2/1, 2/2, and 3/1 on Switch B interact with all the ports of the network except with ports 2/1 and 2/2 on Switch A.
- Ports 2/1 and 2/2 on Switch D do not interact with each other but they interact with all the user ports on Switch A except 3/1, 3/2, and 3/3. They also interact with all the ports on Switch B and Switch C.
- Ports 3/1, 3/2, and 2/1 on Switch C can interact with all the user ports on the network except 3/1, 3/2, and 3/3 on Switch A.

Example Port Mapping Configuration Steps

The following steps provide a quick tutorial to configure the port mapping session shown in the diagram on page 34-6.

1. Configure session 1 on Switch A in the unidirectional mode using the following command:

```
-> port-mapping 1 unidirectional
```

2. Create two port mapping sessions on Switch A using the following commands:

```
-> port-mapping 1 user-port 2/1-2 network-port 1/1-2
-> port-mapping 2 user-port 3/1-3 network-port 1/3
```

3. Enable both the sessions on Switch A using the following commands:

```
-> port-mapping 1 enable
-> port-mapping 2 enable
```

4. Similarly, create and enable a port mapping session 1 on Switch D using the following commands:

```
-> port-mapping 1 user-port 2/1-2 network-port 1/1
-> port-mapping 1 enable
```

Verifying the Port Mapping Configuration

To display information about the port mapping configuration on the switch, use the show commands listed below:

show port-mapping status	Displays the status of one or more port mapping sessions.
show port-mapping	Displays the configuration of one or more port mapping sessions.

For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide.*

35 Configuring Learned Port Security

Learned Port Security (LPS) provides a mechanism for authorizing source learning of MAC addresses on Ethernet ports. Using LPS to control source MAC address learning provides the following benefits:

- A configurable source learning time limit that applies to all LPS ports.
- A configurable limit on the number of MAC addresses (bridged and filtered) allowed on an LPS port.
- Dynamic configuration of a list of authorized source MAC addresses.
- Static configuration of a list of authorized source MAC addresses.
- Three methods for handling unauthorized traffic: administratively disable the LPS port, stop all traffic on the port (port remains up), or only block traffic that violates LPS criteria.

In This Chapter

This chapter provides an over of the LPS feature and describes how to configure LPS parameters through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following information and procedures are included in this chapter:

- "Learned Port Security Defaults" on page 35-1.
- "Sample Learned Port Security Configuration" on page 35-3.
- "Learned Port Security Overview" on page 35-5.
- "Interaction With Other Features" on page 35-9.
- "Configuring Learned Port Security" on page 35-11.
- "Displaying Learned Port Security Information" on page 35-20.

For more information about source MAC address learning, see Chapter 3, "Managing Source Learning."

Learned Port Security Defaults

Parameter Description	Command	Default
LPS status for a port.	**port-security**	disabled

Parameter Description	Command	Default
Number of learned MAC addresses allowed on an LPS port.	**port-security maximum**	1
Maximum number of filtered MAC addresses that the LPS port can learn.	**port-security port max-filtering**	5
Source learning time limit.	**port-security learning-window**	infinity
MAC address range per LPS port.	**port-security mac-range**	00:00:00:00:00:00–ff:ff:ff:ff:ff:ff
LPS port violation mode.	**port-security port violation**	restrict
Number of bridged MAC addresses learned before a trap is sent.	**port-security learn-trap-threshold**	5

Sample Learned Port Security Configuration

This section provides a quick tutorial to perform the following tasks:

- Enabling LPS on a set of switch ports.
- Defining the maximum number of learned MAC addresses allowed on an LPS port.
- Defining the time limit for which source learning is allowed on all LPS ports.
- Selecting a method for handling unauthorized traffic received on an LPS port.

Quick Steps

1 Enable LPS on ports 6 through 8 using the following commands:

```
-> port-security port 1/1/6-8 admin-state enable
```

2 Set the total number of learned MAC addresses allowed on the same ports to 25 using the following command:

```
-> port-security port 1/1/6-8 maximum 25
```

3 Configure the amount of time in which source learning is allowed on all LPS ports to 30 minutes using the following command:

```
-> port-security learning-window 30
```

4 Select **shutdown** for the LPS violation mode using the following command:

```
-> port-security port 1/1/6-8 violation shutdown
```

Note. *Optional*. To verify the LPS port configuration, use the command **show port-security**. For example:

```
 -> show port-security port 1/1/1
Legend: Mac Address: * = address not valid,

        Mac Address: & = duplicate static address,

Port:  1/1/1
 Admin-State      :                ENABLED,
 Operation Mode   :                ENABLED,
 Max MAC bridged  :                      3,
 Trap Threshold   :                      1,
 Violation        :               RESTRICT
 Max MAC filtered :                      5,
 Violating MAC    :                   NULL

            MAC              VLAN       MAC TYPE          OPERATION
-------------------------+--------+-----------------+-----------------
 00:11:22:22:22:22           1       STATIC           bridging
 00:11:22:22:22:21           1       STATIC           bridging
 00:11:22:22:22:21           1       PSEUDO-STATIC    bridging
```

To verify the new source learning time limit value, use the **show port-security learning-window** command. For example:

```
-> show port-security learning-window
Learning-Window           =  30 min,
Convert-to-static         =  DISABLE,
No Aging                  =  DISABLE,
Boot Up                   =  ENABLE,
Learn As Static           =  DISABLE,
Mac Move                  =  DISABLE,
Remaining Learning Window =  1796 sec,
```

Learned Port Security Overview

Learned Port Security (LPS) provides a mechanism for controlling network device access on one or more switch ports. Configurable LPS parameters allow the user to restrict the source learning of host MAC addresses to:

- A specific amount of time in during which source learning is allowed to occur on all LPS ports.
- A maximum number of learned MAC addresses allowed on the port.
- A maximum number of filtered MAC addresses allowed on the port.
- A range of authorized source MAC addresses allowed on the port.

Additional LPS functionality allows the user to specify how the LPS port handles unauthorized traffic. The following options are available for this purpose:

- Block traffic that violates LPS port restrictions; authorized traffic is forwarded on the port.
- Disable learning on the LPS port when unauthorized traffic is received.
- Administratively down the LPS port when unauthorized traffic is received; all traffic is stopped.

LPS functionality is supported on the following port types:

- Fixed
- 802.1Q tagged
- Universal Network Profile (UNP)
- Static and dynamic SAP ports

See the "Interaction With Other Features" on page 35-9.

The following port types are not supported:

- Link aggregate
- Tagged (trunked) link aggregate
- Link aggregate members

LPS Learning Window

The LPS learning window is a configurable amount of time during which source learning of MAC addresses is allowed on LPS ports. This time limit is a global switch value that applies to all LPS-enabled ports; it is not configurable on an individual port basis.

In addition to the source learning time limit, the following learning window options are configurable:

- **Convert dynamically learned MAC addresses to static MAC addresses**. When this option is enabled, all dynamic MAC addresses learned during the learning window time period are converted to static MAC addresses when the learning window closes.
- **Start the learning window when the switch boots up**. When this option is enabled, the learning window time period automatically starts each time the switch restarts.

- **Stop dynamically learned MAC address aging**. When this option is enabled, MAC addresses learned during the learning window time period are learned as pseudo-static MAC addresses. This type of address does not age out or get flushed even after the learning window closes.
- **Allow MAC movement**. When this option is enabled, a pseudo-static MAC address learned on one port can move to another port in the same VLAN without getting dropped.
- **Automatically learn MAC addresses as static MAC addresses**. When this option is enabled, learned MAC addresses are automatically converted to static MAC addresses during the learning window time.

MAC Address Types

There are four types of MAC addresses that are the result of or involved with the LPS port configuration and operation:

- **Static**. A user-configured MAC address on the LPS port.
- **Pseudo Static**. A dynamically learned MAC address that is treated the same as a regular static address (will not age out). However, pseudo-static MAC addresses are not saved in the running configuration of the switch.
- **Dynamic Bridged**. MAC address that are dynamically learned as bridged addresses up to the maximum number of bridged addresses allowed on the LPS port. When this maximum is reached, subsequent addresses are dynamically learned as filtered MAC addresses.
- **Dynamic Filtered**. MAC addresses that are dynamically learned as filtered address up to the maximum number of filtered addresses allowed on the LPS port.

How LPS Authorizes Source MAC Addresses

When a packet is received on a port that has LPS enabled, switch software checks specific criteria to determine if the source MAC address contained in the packet is allowed on the port. The following chart depicts the flow of the MAC address as various LPS rules are applied to determine whether or not the address is learned on the port and the state of the address on that port (bridged or filtered)

:

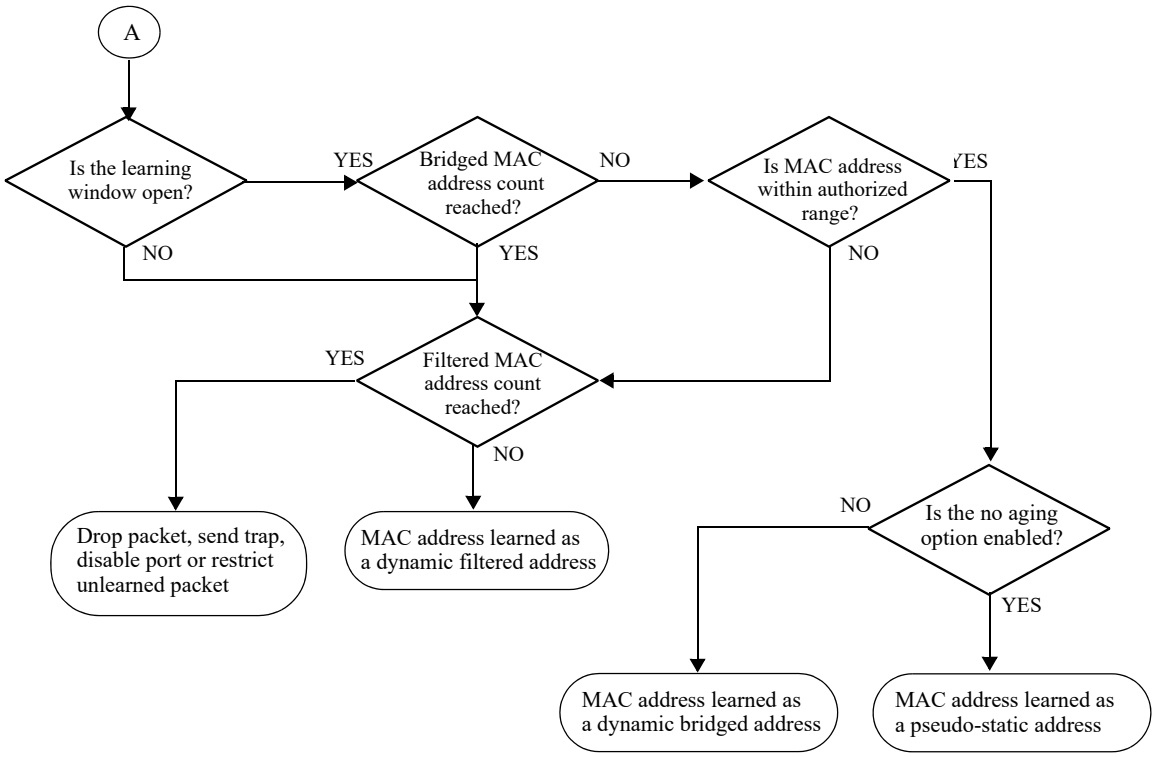


Figure 35-1 : How LPS Authorizes Source MAC Addresses

MAC Address Behavior on LPS Ports

The following table shows how LPS MAC addresses are treated when specific switch or LPS actions are taken:

Action	Static	Pseudo-Static	Dynamic Bridged	Dynamic Filtered
LPS port removed	Flushed	Flushed	Flushed	Flushed
Write memory	Written	Not written	Not written	Not written
Convert to static MAC	No change	Converted	Converted	No change
LPS admin disable	No change	No change	Flushed	Flushed
Enable after disable	No change	No change	Flushed	Flushed
LPS admin locked	No change	No change	No change	No change
Enable after locked	No change	No change	No change	No change
Aging bridged MAC	None	None	Aged entry removed	Flushed
Aging filtered MAC	None	None	No change	Aged entry removed
Remove static MAC	Entry removed	No change	Can learn one	Flushed
Remove pseudo-static MAC	No change	Entry removed	Can learn one	Flushed
Remove dynamic bridged	No change	No change	Can learn one	Flushed
Remove dynamic filtered	No change	No change	No change	Can learn one
Modify trap threshold	No change	No change	No change	No change

Action	Static	Pseudo-Static	Dynamic Bridged	Dynamic Filtered
Modify violation	No change	No change	No change	No change
Increase bridged maximum	No change	No change	Can learn more	Flushed
Decrease bridged maximum	No change	No change	Flushed	Flushed
Increase filtered maximum	No change	No change	No change	Can learn more
Decrease filtered maximum	No change	No change	No change	Flushed
Change MAC range	No change	No change	Flushed	Flushed
Changes During the Learning Window Time Period				
Disable boot up	No change	None	None	Can learn
Enable boot up	No change	None	Can learn	Can learn
Enable no aging	No change	Can learn	Don’t learn	Can learn
Disable no aging	No change	Don’t learn	Can learn	Can learn
Enable convert to static	No change	Convert at timeout	Convert at timeout	No change
Disable convert to static	No change	No change	No change	No change
Enable learn as static	No change	Convert to static	Convert to static	No change
Disable learn as static	No change	No change	No change	No change
Enable MAC movement	No change	MAC move allowed	No change	No change
Disable MAC movement	No change	No change	No change	No change

Note. If the LPS learning window time period is set to infinity, enabling the convert to static function is not allowed. If convert to static is already enabled and the learning window time is changed to infinity, convert to static is automatically disabled.

Dynamic Configuration of Authorized MAC Addresses

When LPS is configured on a switch port, the learning of source MAC addresses is initiated. An entry containing the address and the port that learns the MAC address is made in an LPS database table. This entry is used as a criteria for authorizing future traffic from the source MAC address on that same port. In other words, the learned MAC addresses are authorized to send traffic through the LPS port.

For example, if the source MAC address 00:da:95:00:59:0c is received on port 2/10 and meets the LPS restrictions defined for that port, then this address and its port are recorded in the LPS table. All traffic that is received on port 2/10 is compared to the 00:da:95:00:59:0c entry. If any traffic received on this port consists of packets that do not contain a matching source address, the packets are then subject to the LPS source learning time limit window and the criteria for maximum number of addresses allowed.

Static Configuration of Authorized MAC Addresses

It is also possible to statically configure authorized source MAC address entries into the LPS table. This type of entry behaves the same way as dynamically configured entries providing authorized port access to traffic that contains a matching source MAC address.

Static source MAC address entries, however, take precedence over dynamically learned entries. For example, if there are 2 static MAC address entries configured for port 1/2/1 and the maximum number allowed on port 1/2/1 is 10, then only 8 dynamically learned MAC addresses are allowed on this port.

There are three ways to configure a static source MAC address entry in the LPS table:

- Use the LPS **port-security mac** command to manually configure a static MAC address for one or more LPS ports.
- Use the LPS learning window **convert-to-static, no-aging**, or **learn-as-static** options (see "Configuring the LPS Learning Window" on page 35-12 for more information).
- Use the LPS **port-security convert-to-static** command to manually convert all dynamic addresses on a specific port to static MAC addresses.

Note. Statically configured authorized MAC addresses are displayed permanently in the MAC address table for the specified LPS port; they are not learned on any other port in the same VLAN.

Understanding the LPS Table

The LPS database table is separate from the source learning MAC address table. However, when a MAC is authorized for learning on an LPS port, an entry is made in the MAC address table in the same manner as if it was learned on a non-LPS port (see Chapter 3, "Managing Source Learning," for more information).

In addition to dynamic and configured source MAC address entries, the LPS table also provides the following information for each eligible LPS port:

- The LPS status for the port; enabled or disabled.
- The maximum number of MAC addresses allowed on the port.
- The maximum number of MAC addresses that can be filtered on the port.
- The violation mode selected for the port; restrict, discard, or shutdown.
- Statically configured MAC addresses and MAC address ranges.
- All MAC addresses learned on the port.
- The management status for the MAC address entry; configured or dynamic.

If the LPS port is shut down or the network device is disconnected from the port, the LPS table entries and the source learning MAC address table entries for the port are automatically cleared.

To view the contents of the LPS table, use the **show port-security** command. Refer to the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about this command.

Interaction With Other Features

This section contains important information about how the Learned Port Security (LPS) functionality interacts with other OmniSwitch features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

Access Guardian

LPS is one of the Access Guardian security functions that helps to ensure that only certain devices are allowed to connect to the switch. The LPS functionality is used to control which MAC addresses are allowed on a switch port.

Universal Network Profile (UNP)

Access Guardian is configured and applied through the framework of the UNP feature. UNP is enabled on switch ports to activate Access Guardian functionality that is used to authenticate and classify users into UNP profiles. LPS is supported on UNP-enabled ports.

- When both of these features are enabled on the same port, UNP first authenticates and classifies any MAC addresses received, then LPS rules are applied. If a MAC address violates any of the LPS rules for the port, the address may get filtered or the port violated even if UNP initially determined the address was valid. In other words, LPS rules take precedence over UNP to determine if a MAC address is bridged or filtered on the port.
- When UNP is enabled on any one LPS port, the LPS learning window parameter options are not supported on all LPS-enabled ports. This is because the learning window configuration is global and applies to all LPS ports.

For more information about LPS on UNP ports, see Chapter 47, “Learned Port Security Commands.”

Shortest Path Bridging (SPB)

LPS on Static Service Access Ports (SAP)

A static SAP and its mapping to the service must be pre-configured using the Service Manager CLI **service sap** before LPS can be configured on the respective SAP.

```
-> service 100 sap port 1/1/1:10
-> port-security sap 1/1/1:10 maximum 1000
```

LPS on Dynamic SAPs

If LPS is to be configured on a dynamic SAP, the dynamic SAP (and its service mapping) on the UNP port, must pre-exist by creating it upfront through persistent-profile configuration on the UNP access port or linkagg. Statically assigning a UNP service profile to a UNP port creates a persistent Service Access Point (SAP) that will not age out when there is no activity on the port. This solution is particularly useful for access to silent devices in the UNP service domain.

Configuring Learned Port Security

This section describes how to use Command Line Interface (CLI) command to configure Learned Port Security (LPS) on a switch. See the "Sample Learned Port Security Configuration" on page 35-3 for a brief tutorial on configuring LPS.

Configuring LPS involes the following procedures:

- Enabling LPS for one or more switch ports. This procedure is described in "Configuring the LPS Port Administrative Status" on page 35-11.
- Configuring the source learning time window during which MAC addresses are learned. This procedure is described in "Configuring the LPS Learning Window" on page 35-12.
- Configuring the maximum number of bridged MAC addresses allowed on an LPS port. This procedure is described in "Configuring the Number of Bridged MAC Addresses Allowed" on page 35-16.
- Configuring the maximum number of filtered MAC addresses allowed on an LPS port. This procedure is describe in "Configuring the Number of Filtered MAC Addresses Allowed" on page 35-17
- Configuring a range of authorized MAC addresses allowed on an LPS port. This procedure is described in "Configuring an Authorized MAC Address Range" on page 35-17.
- Specifying whether or not an LPS port shuts down all traffic or only restricts traffic when an unauthorized MAC address is received on the port. This procedure is described in "Selecting the Security Violation Mode" on page 35-19.

Configuring the LPS Port Administrative Status

The **port-security** command is used to configure the administrative status of LPS on a port using one of the following three parameter options:

enable	Enables LPS functionality on the port. When LPS is enabled: • All MAC addresses are cleared. • The LPS configuration is applied to source learning on the port. • The port can go into a shutdown, restricted, or discard state (based on the configured violation mode) when unauthorized addresses are received on the port.
disable	Disables LPS functionality on the port. When LPS is disabled: • All filtered and bridged MAC addresses are cleared. • Pseudo-static and static addresses remain in a forwarding state. • The static MAC configuration is retained. • The LPS configuration is retained but not applied. • Learning on the port is wide open; not restricted by LPS.
locked	Disables all learning on the port. When LPS is locked: • Existing MAC addresses are retained. • No additional learning is allowed. • Static MAC addresses are still allowed.

Enabling LPS Functionality on a Port

By default, LPS is disabled on all switch ports. To enable LPS on a port, use the **port-security** command. with the **admin-state enable** parameter. For example, the following command enables LPS on port 1/4:

```
-> port-security port 4/1/1 admin-state enable
```

To enable LPS on multiple ports, specify a range of ports. For example:

```
-> port-security port 4/1/1-5 admin-state enable
-> port-security port 5/1/12-20 admin-state enable
```

Note. When LPS is enabled on an active port, all MAC addresses learned on that port prior to the time LPS was enabled are cleared from the source learning MAC address table.

Disabling LPS Functionality on a Port

To disable LPS on a port, use the **port-security** command with the **admin-state disable** parameter. For example, the following command disables LPS on a range of ports:

```
-> port-security 5/1/21-24 admin-state disable
```

To disable all the LPS ports on a chassis, use the **port-security chassis admin-state** command, as shown:

```
-> port-security chassis admin-state disable
```

When LPS is disabled on a port, the MAC address entries for that port are retained in the LPS table. The next time LPS is enabled on the port, the same LPS table entries become active again. If there is a switch reboot before the switch configuration is saved, however, dynamic MAC address entries are discarded from the table.

Locking the LPS Port

To lock the LPS port, use the **port-security** command with the **admin-state locked** parameter. For example, the following command locks port 5/1/21:

```
-> port-security 5/1/21 admin-state locked
```

When the LPS port is locked, all learning on the port is stopped.

Removing the LPS Configuration from the Port

Use the **no** form of the **port-security** command to remove the LPS configuration and clear all entries (configured and dynamic) in the LPS table for the specified port. For example:

```
-> no port-security port 5/1/10
```

After LPS is removed, all the dynamic and static MAC addresses are flushed and unrestricted learning of new MAC addresses is enabled.

Configuring the LPS Learning Window

By default, the LPS source learning window time limit is set to infinity. This means that there is no limit on the amount of time during which MAC addresses are learned on all LPS ports. To limit the amount of time that source learning is allowed on LPS ports, use the **port-security learning-window** command.

During the time the learning window is open, source MAC addresses that comply with LPS port restrictions are authorized for source learning on the related LPS port. The following actions trigger the start of the learning window timer:

- Using the **port-security learning-window** command. Each time this command is issued, the timer restarts even if a current window is still open.

- A switch reboot with the **port-security learning-window** command entry saved in the **vcboot.cfg** file. When this command is used to configure the learning window time and related options for the switch, use the **write memory** command to ensure the command is saved in the **vcboot.cfg** file.

The LPS learning window time limit is a switch-wide parameter that applies to all LPS-enabled ports, not just one or a group of LPS ports. The following command example sets the time limit value to 30 minutes:

```
-> port-security learning-window 30
```

Note. When the time limit value expires, source learning of any new dynamic bridged MAC addresses is stopped on all LPS ports, even if the number of bridged addresses learned does not exceed the maximum allowed. However, after the window has closed, the switch will continue to learn dynamic filtered MAC addresses until the maximum number of filtered addresses allowed is reached.

Setting the LPS learning window time value to zero (the default) configures an infinite learning window for LPS ports. For example:

```
-> port-security learning-window 0
```

Use the **show port-security learning-window** command to determine the current settings for the LPS learning window.

Configuring Learning Window Parameters

In addition to specifying the duration of the LPS learning window, the **port-security learning-window** command provides the following parameters for configuring additional learning window options:

convert-to-static	Specifies whether or not learned dynamic bridged MAC addresses are converted to static MAC addresses when the learning window closes. See "Converting Dynamic MAC Addresses to Static MAC Addresses" on page 35-13.
no-aging	Specifies whether or not learned dynamic MAC addresses can age out. See "Configuring the MAC Address Aging Status" on page 35-14.
mac-move	Specifies whether or not a pseudo-static MAC address learned on one LPS port can move to another LPS port in the same VLAN without getting dropped. This option is used together with the **no-aging** option. See "Configuring the MAC Movement Status" on page 35-15.
learn-as-static	Specifies whether or not learned dynamic bridged MAC addresses are automatically converted to static MAC addresses during the learning window time frame. This option and the **no-aging** option are mutually exclusive. See "Learning MAC Addresses as Static MAC Addresses" on page 35-15.
boot-up	Specifies whether or not the learning window timer will automatically start each time the switch restarts. See "Starting the Learning Window at Boot Up" on page 35-16.

Converting Dynamic MAC Addresses to Static MAC Addresses

When the learning window time expires, all the dynamic MAC addresses learned on the LPS ports start to age out. The **convert-to-static** parameter option of the **port-security learning-window** command is used

to specify whether or not these MAC addresses are converted to static addresses when the learning window time period ends.

Note. The number of converted static MAC addresses cannot exceed the maximum number of MAC addresses allowed on the LPS ports.

By default, converting dynamic MAC addresses to static MAC addresses is disabled. To enable this option for the learning window, use the following command:

```
-> port-security learning-window 30 convert-to-static enable
```

If the LPS learning window time is set to zero (infinity), enabling the **convert-to-static** option is not allowed. For example:

```
-> port-security learning-window 0 convert-to-static enable
ERROR: Convert-to-static cannot be configured along with infinite learning-
window
```

The following command disables this option for the learning window:

```
-> port-security learning-window 30 convert-to-static disable
```

If the LPS learning window is set to a specific time and the **convert-to-static** option is enabled, the **convert-to-static** option is automatically disabled when the learning window time is set to zero. For example:

```
-> port-security learning-window 10 convert-to-static enable

-> show port-security learning-window
Learning-Window           =  10 min,
Convert-to-static         =  ENABLE,
No Aging                  =  DISABLE,
Boot Up                   =  ENABLE,
Learn As Static           =  DISABLE,
Mac Move                  =  DISABLE,
Remaining Learning Window =  594 sec,

-> port-security learning-window 0

-> show port-security learning-window
Learning-Window           =  INFINITY,
Convert-to-static         =  DISABLE,
No Aging                  =  DISABLE,
Boot Up                   =  ENABLE,
Learn As Static           =  DISABLE,
Mac Move                  =  DISABLE
```

Configuring the MAC Address Aging Status

During the learning window time period, dynamically learned MAC addresses may age out before the learning window time is up. To prevent this from happening, use the **no-aging enable** parameter option with the **port-security learning-window** command.

When this option is enabled, all dynamic bridged MAC addresses are learned as pseudo-static MAC addresses. This type of address is treated as a regular statically configured address and will not age out, even after the learning window closes. However, pseudo-static MAC addresses are not saved in the switch configuration and MAC movement is not allowed for this type of MAC address.

The no MAC address aging option is best used in combination with the option that converts dynamic addresses to static addresses. Enabling both of these options ensures that no learned MAC addresses will age out before or after the learning window closes..

Note. The no MAC address aging option and the learn MAC addresses as static MAC addresses option are mutually exclusive. If both are enabled, then the learn MAC addresses as static MAC addresses option takes precedence.

By default, the no MAC address aging status is disabled. To enable this option for the learning window, use the following command:

```
-> port-security learning-window no-aging enable
```

To disable this option for the learning window, use the following command:

```
-> port-security learning-window no-aging disable
```

Configuring the MAC Movement Status

MAC addresses are learned as pseudo-static MAC addresses when the **no-aging** option is enabled for the LPS learning window. If a duplicate pseudo-static MAC address is seen on another port in the same VLAN, the MAC address is dropped. To prevent this from happening, use the **mac-move enable** parameter option with the **port-security learning-window** command.

When MAC movement is enabled, a pseudo-static MAC address can move to another port within the same VLAN without getting dropped. After the move has occurred, the switch configuration is updated to reflect the new port association for the pseudo-static MAC address. No information about the original port association is retained.

By default, the MAC movement option is disabled. To enable this option for the learning window, use the following command:

```
-> port-security learning-window 30 mac-move enable
```

The following command disables this option for the learning window:

```
-> port-security learning-window 30 mac-move disable
```

Learning MAC Addresses as Static MAC Addresses

If a switch reboots during an active LPS learning window, MAC addresses learned up to that point are lost. To prevent this from happening, use the **learn-as-static enable** parameter option with the **port-security learning-window** command.

When this option is enabled, all MAC addresses are automatically learned as static MAC addresses during the learning window time period. These static MAC addresses are saved to the switch configuration.

Note. The **learn-as-static** and **no-aging** learning window options are mutually exclusive. If both options are enabled, then the **learn-as-static** option takes precedence.

By default, the option to learn MAC addresses as static MAC addresses is disabled. To enable this option for the learning window, use the following command:

```
-> port-security learning-window 30 learn-as-static enable
```

The following command disables this option for the learning window:

```
-> port-security learning-window 30 learn-as-static disable
```

Starting the Learning Window at Boot Up

By default, the **boot-up** option is enabled when the learning window time is configured. This option specifies that whenever the switch reboots, the learning window time period will automatically restart at the time the reboot occurs.

To disable this functionality, use the **boot-up disable** parameter with the **port-security learning-window** command. For example:

```
-> port-security learning-window boot-up disable
```

To enable this functionality, use the **boot-up enable** parameter with the **port-security learning-window** command. For example:

```
-> port-security learning-window boot-up disable
```

Note. After the **boot-up** option is enabled (either by default or explicitly configured), perform the **write memory** command to save the **port-security learning-window** command to the switch configuration file. This will ensure that the learning window will automatically start when the switch reboots.

Configuring the Number of Bridged MAC Addresses Allowed

To configure the number of bridged MAC addresses allowed on an LPS port, use the **port-security maximum** command. For example, the following command sets the maximum number of MAC addresses learned on port 10 to 75:

```
-> port-security port 6/1/10 maximum 75
```

To specify a maximum number of MAC addresses allowed for multiple ports, specify a range of ports. For example:

```
-> port-security port 1/1/10-15 maximum 10
-> port-security port 2/1/1-5 maximum 25
```

If there are 10 configured authorized MAC addresses for an LPS port and the maximum number of addresses allowed is set to 15, then only 5 dynamically learned MAC address are allowed on this port.

If the maximum number of MAC addresses allowed is reached before the switch LPS time limit expires, then all source learning of dynamic *and* configured bridged MAC addresses is stopped on the LPS port. However, the switch will continue to learn subsequent addresses as filtered until the maximum number of filtered MAC addresses allowed on the port is reached.

Configuring the Trap Threshold for Bridged MAC Addresses

The LPS trap threshold value determines how many bridged MAC addresses the port must learn before a trap is sent. Once this value is reached, a trap is sent for every MAC learned thereafter.

By default, when one bridged MAC addresses is learned on an LPS port, the switch sends a trap. To change the trap threshold value, use the **port-security learn-trap-threshold** command. For example:

```
-> port-security port learn-trap-threshold 10
```

Sending a trap when this threshold is reached provides notification of newly learned bridged MAC addresses. Trap contents includes identifying information about the MAC, such as the address itself, the corresponding IP address, switch identification, and the slot/port number on which the MAC was learned.

Configuring the Number of Filtered MAC Addresses Allowed

To configure the number of filtered MAC addresses allowed on an LPS port, use the **port-security port max-filtering** command. For example, the following command sets the maximum number of filtered MAC addresses learned on port 9 to 18:

```
-> port-security port 5/1/9 max-filtering 18
```

To specify a maximum number of filtered MAC addresses learned on multiple ports, specify a range of ports or multiple slots. For example:

```
-> port-security port 5/1/9-15 max-filtering 10
-> port-security port 1/1/1-5 max-filtering 25
```

If the maximum number of filtered MAC addresses allowed is reached:

- The violation mode configured for the LPS port is applied (see "Selecting the Security Violation Mode" on page 35-19 for more information).
- An SNMP trap is generated.
- An event is entered into the switch log.

Configuring an Authorized MAC Address Range

By default, each LPS port is set to a range of 00:00:00:00:00:00–ff:ff:ff:ff:ff:ff, which includes all MAC addresses. If this default is not changed, then addresses received on LPS ports are subject only to the learning window time and restrictions on the maximum number of MAC addresses allowed for the port.

All MAC addresses that fall within the default or a specific configured range of addresses are dynamically learned as bridged MAC addresses (up to the maximum of bridged addresses allowed). If a MAC address falls outside of the specified range, the address is dynamically learned as a filtered MAC address (up to the maximum of filtered addresses allowed).

To configure a source MAC address range for an LPS port, use the **port-security mac-range** command. For example, the following command configures a MAC address range for port 4:

```
-> port-security port 1/1/4 mac-range low 00:20:da:00:00:10 high
00:20:da:00:00:50
```

The following command examples configure a MAC address range for a range of ports:

```
-> port-security port 1/4/1-5 mac-range low 00:20:da:00:00:10 high
00:20:da:00:00:50
-> port-security port 1/2/1-4 mac-range low 00:20:d0:59:0c:9a high
00:20:d0:59:0c:9f
```

Multiple MAC address ranges (up to eight) can be configured for a port.

The following commands provide an example of configuring multiple MAC ranges for the same port:

```
-> port-security port 1/1/5 mac-range low 00:01:01:22:22:56 high
00:01:01:22:22:67
-> port-security port 1/1/5 mac-range low 00:01:01:22:33:56 high
00:01:01:22:33:67
-> port-security port 1/1/5 mac-range low 00:01:01:22:44:56 high
00:01:01:22:44:67
-> port-security port 1/1/5 mac-range low 00:01:22:22:11:56 high
00:01:22:22:11:67
```

```
-> port-security port 1/1/5 mac-range low 00:01:22:22:22:56 high
00:01:22:22:22:67
-> port-security port 1/1/5 mac-range low 00:01:22:22:33:56 high
00:01:22:22:33:67
-> port-security port 1/1/5 mac-range low 00:01:22:22:44:56 high
00:01:22:22:44:67
-> port-security port 1/1/5 mac-range low 00:01:22:22:55:56 high
00:01:22:22:55:67
```

Note. When a new MAC range is configured, the default MAC range is replaced by the configured MAC range. The default MAC range is automatically applied when all the configured MAC ranges for the port are deleted.

To delete a configured MAC address range, use the **no** form of the **port-security mac-range** command. For example:

```
-> no port-security port 1/1/5 mac-range low 00:01:01:22:44:56
```

Modifying an existing MAC address range is allowed only if the low-end MAC address is *not* changed and the defined new range does not overlap with the existing MAC range. For example, the following command changes only the high-end MAC address 00:01:22:22:55:67 of an existing range to 00:01:22:22:55:70:

```
-> port-security port 1/1/5 mac-range low 00:01:22:22:55:56 high
00:01:22:22:55:70
```

To modify the low-end MAC address, the existing range must be deleted before adding the new range. For example, the following commands change the low-end MAC address 00:01:22:22:55:56 to 00:01:22:22:55:56:

```
-> no port-security port 1/1/5 mac-range low 00:01:22:22:55:56 high
00:01:22:22:55:70

-> port-security port 1/1/5 mac-range low 00:01:22:22:55:60 high
00:01:22:22:55:70
```

Note.

- When modifying a MAC range, the new range must match or accommodate any existing static MACs on the port, else an error will be thrown indicating some static MACs exist on the port that fall outside the new/resultant MAC range being configured. (Note: It is required to flush such static MACs on the port, if user needs to configure the new MAC range, which was not accommodating the static MACs)
- When the MAC range size is increased, all the dynamic filtering MACs on the port would be flushed.
- When the MAC range size is reduced, any existing dynamic forwarding MACs learned on the port would be flushed if they fall outside any MAC ranges configured on the port at that point of time.
- All the dynamic filtering MACs learned on the port would be flushed.

To view the configured MAC range for the port, use the **show port-security mac-range** command.

Refer to the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about this command.

Selecting the Security Violation Mode

The **port-security port violation** command configures the violation mode (restrict, discard, or shutdown) that is applied to an LPS port when the maximum number of bridged and filtered addresses allowed on the port is reached. Use the following table to determine how each violation mode is applied and which actions or events will clear the violation state and return the port to normal operation:

Mode (Parameter)	Violation Mode Description	Violation Recovery
restrict	Port remains up but unauthorized MAC addresses are blocked. All other packets that contain an authorized source MAC address are allowed to continue forwarding on the port.	• Bridge and filtered MAC addresses age out. • MAC addresses are flushed. • Use **clear violation** command. • Link down/up event. • LPS port is removed.
discard	Port remains up but all traffic received on the port is discarded. Dynamically learned MAC addresses are flushed.	• Use **clear violation** command. • Link down/up event. • LPS port is removed.
shutdown	Port is administratively disabled. All traffic is stopped at the port; no traffic is forwarded.	• Use **clear violation** command. • Link down/up event. • LPS port is removed.

Note. Unauthorized source MAC addresses are not learned in the LPS table but are still recorded in the source learning MAC address table with a filtered operational status. This allows the user to view MAC addresses that were attempting unauthorized access to the LPS port.

By default, the security violation mode for an LPS port is set to **restrict**. To configure the security violation mode for an LPS port, enter **port-security** followed by the *slot/port* designation of the port, then **violation** followed by **restrict**, **discard**, or **shutdown**. For example, the following command selects the shutdown mode for port 1:

```
-> port-security port 4/1/1 violation shutdown
```

To configure the security violation mode for multiple LPS ports, specify a range of ports or multiple slots. For example:

```
-> port-security port 4/1/1-10 violation shutdown
-> port-security port 4/1/10-15 violation restrict
```

To verify the details about LPS violations, use the **show violation** command. For example:

```
-> show violation

Port    Source     Action              Reason           Timer
-------+----------+-------------------+----------------+--------
  1/1    src lrn    simulated down      lps shutdown     0
  1/2    qos        simulated down      policy           0
  2      udld       admin down          udld             0
```

To clear all the LPS violation information use the **interfaces hybrid-mode** command.

Displaying Learned Port Security Information

To display Learned Port Security (LPS) port and table information, use the **show** commands listed below:

show port-security	Displays the LPS configuration and table entries.
show port-security learning-window	Displays the amount of time during which source learning can occur on all LPS ports.
show violation	Displays the address violations that occur on ports with LPS restrictions.

For more information about the resulting display from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. An example of the output for the **show port-security, show port-security learning-window,** and **show violation** commands is also given in "Sample Learned Port Security Configuration" on page 35-3.

36 Diagnosing Switch Problems

Several tools are available for diagnosing problems that occur with the switch. These tools include:

- Port Mirroring
- Port Monitoring
- sFlow
- Remote Monitoring (RMON) probes
- Switch Health Monitoring

Port mirroring copies all incoming and outgoing traffic from configured mirror ports to a second mirroring Ethernet port, where it can be monitored with a Remote Network Monitoring (RMON) probe or network analysis device without disrupting traffic flow on the mirrored port. The port monitoring feature allows you to examine packets to and from a specific Ethernet port. sFlow is used for measuring high speed switched network traffic. It is also used for collecting, storing, and analyzing the traffic data. Switch Health monitoring software checks previously configured threshold levels for the switch's consumable resources, and notifies the Network Monitoring Station (NMS) if those limits are violated.

In This Chapter

This chapter describes port mirroring, port monitoring, remote monitoring (RMON) probes, sFlow, and switch health features and explains how to configure the same through the Command Line Interface (CLI).

Configuration procedures described in this chapter include:

- Creating or Deleting a Port Mirroring Session—see "Creating a Mirroring Session" on page 36-14 or "Deleting A Mirroring Session" on page 36-17.
- Protection from Spanning Tree changes (Port Mirroring)—see "Unblocking Ports (Protection from Spanning Tree)" on page 36-15.
- Enabling or Disabling Port Mirroring Status—see "Enabling or Disabling Mirroring Status" on page 36-15 or "Disabling a Mirroring Session (Disabling Mirroring Status)" on page 36-15.
- Configuring Port Mirroring Direction—see "Configuring Port Mirroring Direction" on page 36-16.
- Enabling or Disabling a Port Mirroring Session—see "Enabling or Disabling a Port Mirroring Session (Shorthand)" on page 36-17.
- Configuring RPMIR Loopback Session—see "Configuring Remote Port Mirroring (RPMIR) Loopback Session for Multiple Destination Ports" on page 36-20

- Setting Port Mirroring Traffic Priority—see "Setting Port Mirroring Traffic Priority" on page 36-22
- Configuring a Port Monitoring Session—see "Configuring a Port Monitoring Session" on page 36-23.
- Enabling a Port Monitoring Session—see "Enabling a Port Monitoring Session" on page 36-24.
- Disabling a Port Monitoring Session—see "Disabling a Port Monitoring Session" on page 36-24.
- Deleting a Port Monitoring Session—see "Deleting a Port Monitoring Session" on page 36-24.
- Pausing a Port Monitoring Session—see "Pausing a Port Monitoring Session" on page 36-25.
- Configuring the persistence of a Port Monitoring Session—see "Configuring Port Monitoring Session Persistence" on page 36-25.
- Configuring a Port Monitoring data file—see "Configuring a Port Monitoring Data File" on page 36-25.
- Configuring a Port Monitoring direction—see "Configuring Port Monitoring Direction" on page 36-26.
- Configuring capture-type—see "Configuring the Capture Type" on page 36-27.
- Displaying Port Monitoring Status and Data—see "Displaying Port Monitoring Status and Data" on page 36-27.
- Configuring a sFlow Session—see "Configuring a sFlow Session" on page 36-29.
- Configuring a Fixed Primary Address—see "Configuring a Fixed Primary Address" on page 36-30.
- Displaying a sFlow Receiver—see "Displaying a sFlow Receiver" on page 36-30.
- Displaying a sFlow Sampler—see "Displaying a sFlow Sampler" on page 36-31.
- Displaying a sFlow Poller—see "Displaying a sFlow Poller" on page 36-31.
- Displaying a sFlow Agent—see "Displaying a sFlow Agent" on page 36-31.
- Deleting a sFlow Session—see "Deleting a sFlow Session" on page 36-32.
- Enabling or Disabling RMON Probes—see "Enabling or Disabling RMON Probes" on page 36-35.
- Configuring Resource Threshold Limits (Switch Health)—see "Configuring Resource Thresholds" on page 36-41.
- Configuring Sampling Intervals—see "Configuring Sampling Intervals" on page 36-43.

For information about additional Diagnostics features such as Switch Logging and System Debugging/ Memory Management commands, see Chapter 38, "Using Switch Logging."

Port Mirroring Overview

The following sections detail the specifications, defaults, and quick set up steps for the port mirroring feature. Detailed procedures are found in "Port Mirroring" on page 36-10.

Port Mirroring Defaults

The following table shows port mirroring default values.

Parameter Description	CLI Command	Default Value/Comments
Mirroring Session Creation	**port-mirroring source destination**	No Mirroring Sessions Configured
Protection from Spanning Tree (Spanning Tree Disable)	**port-mirroring source destination**	Spanning Tree Enabled
Mirroring Status Configuration	**port-mirroring source destination**	Enabled
Mirroring Session Configuration	**port-mirroring**	Enabled
Mirroring Session Deletion	**port-mirroring**	No Mirroring Sessions Configured

Quick Steps for Configuring Port Mirroring

1 Create a port mirroring session. Be sure to specify the port mirroring session ID, source *(mirrored)* and destination *(mirroring)* chassis/slot/port, and unblocked VLAN ID (*optional*—protects the mirroring session from changes in Spanning Tree if the mirroring port monitors mirrored traffic on an RMON probe belonging to a different VLAN). For example:

```
-> port-mirroring 6 source 1/2/3-9 destination 1/2/10 unblocked-vlan 7
```

Note. *Optional.* To verify the port mirroring configuration, enter **show port-mirroring status** followed by the port mirroring session ID number. The display is similar to the one shown below:

```
-> show port-mirroring status 6
Session    Mirror        Mirror      Unblocked   RPMIR     Config    Oper
           Destination    Direction     Vlan       Vlan     Status    Status
----------+-----------+----------------+--------+---------+-------+---------
     1.     1/1/11       bidirectional  NONE      NONE      Enable    on
----------+-----------+----------------+--------+---------+---------+---------
         Mirror
         Source
----------+-----------+--------------+----------+---------+----------+---------
     1.      1/1/2       bidirectional     -         -      Enable    On
     1.      1/1/3       bidirectional     -         -      Enable    On
     1.      1/1/4       bidirectional     -         -      Enable    On
     1.      1/1/5       bidirectional     -         -      Enable    On
```

Note. For more information about this command, see "Displaying Port Mirroring Status" on page 36-17 or the "Port Mirroring and Monitoring Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide.*

Port Monitoring Overview

The following sections detail the specifications, defaults, and quick set up steps for the port mirroring feature. Detailed procedures are found in "Port Monitoring" on page 36-23.

Port Monitoring Defaults

The following table shows port mirroring default values.

Parameter Description	CLI Command	Default Value/Comments
Monitoring Session Creation	**port-monitoring source**	No Monitoring Sessions Configured
Monitoring Status	**port-monitoring source**	Disabled
Monitoring Session Configuration	**port-monitoring source**	Disabled
Port Monitoring Direction	**port-monitoring source**	Bidirectional
Data File Creation	**port-monitoring source**	Enabled
Data File Size	**port-monitoring source**	64K
File Overwriting	**port-monitoring source**	Enabled
Time before session is deleted	**port-monitoring source**	0 seconds
Capture-type	**port-monitoring source**	brief
Set the priority for mirroring traffic	**port-mirroring traffic-priority**	normal

Quick Steps for Configuring Port Monitoring

1 To create a port monitoring session, use the **port-monitoring source** command by entering **port monitoring**, followed by the port monitoring session ID, **source**, and the chassis/slot/port number to be monitored. For example:

```
-> port-monitoring 6 source port 1/2/3
```

2 Enable the port monitoring session by entering **port-monitoring**, followed by the port monitoring session ID, **source**, the chassis/slot/port number to be monitored, and **enable**. For example:

```
-> port-monitoring 6 source port 1/2/3 enable
```

3 *Optional*. Configure optional parameters. For example, to create a file called "monitor1" for port monitoring session 6 on port 1/2/3, enter:

```
-> port-monitoring 6 source port 1/2/3 file monitor1
```

Note. *Optional.* To verify the port monitoring configuration, enter **show port-monitoring status**, followed by the port monitoring session ID number. The display is similar to the one shown below:

```
-> show port-monitoring status
 Sess      Mon.   Mon          Over   Oper.  Admin  Capt.  Max.   File
           Src    Dir          write  Stat   Stat   Type   Size   Name
----------+------+------------+------+------+-----+------+------+-------
 6.        1/2/3   Bidirectional ON     ON     ON     brief
```

For more information about this command, see "Port Monitoring" on page 36-23 or the "Port Mirroring and Monitoring Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide.*

sFlow Overview

The following sections detail the specifications, defaults, and quick set up steps for the sFlow feature. Detailed procedures are found in "sFlow" on page 36-28.

sFlow Defaults

The following table shows sFlow default values:

Parameter Description	CLI Command	Default Value/Comments
Receiver Name	**sflow receiver**	Empty
Receiver Timeout Value	**sflow receiver**	No Timeout
Receiver IP Address	**sflow receiver**	0.0.0.0
Receiver Data File Size	**sflow receiver**	1400 Bytes
Receiver Version Number	**sflow receiver**	5
Receiver Destination Port	**sflow receiver**	6343
Sampler Rate	**sflow sampler**	0
Sample Header Size	**sflow sampler**	128 Bytes
Poller Interval Value	**sflow poller**	5 seconds

Quick Steps for Configuring sFlow

Follow the steps below to create an sFlow receiver session.

1 To create a sFlow receiver session, use the **sflow receiver** command by entering **sflow receiver**, followed by the receiver index, name, and the IP address. For example:

```
-> sflow receiver 1 name Golden address 198.206.181.3
```

2 *Optional*. Configure optional parameters. For example, to specify the timeout value "65535" for sFlow receiver session on address 198.206.181.3, enter:

```
-> sflow receiver 1 name Golden address 198.206.181.3 timeout 65535
```

Note. *Optional.* To verify the sFlow receiver configuration, enter **show sflow receiver**, followed by the sFlow receiver index. The display is similar to the one shown below:

```
-> show sflow receiver

Receiver 1
Name        = Golden
Address     = IP_V4 198.206.181.3
UDP Port    = 6343
Timeout     = 65535
Packet Size= 1400
DatagramVer= 5
```

For more information about this command, see "sFlow" on page 36-28 or the "sFlow Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide.*

Follow the steps below to create a sFlow sampler session.

1 To create a sFlow sampler session, use the **sflow sampler** command by entering **sflow sampler**, followed by the instance ID, port list, receiver, and the rate. For example:

```
-> sflow sampler 1 port 2/1/1-5 receiver 1 rate 2048
```

2 *Optional*. Configure optional parameters. For example, to specify the sample-hdr-size value "128" for sFlow sampler instance 1 on ports 2/1/1-5, enter:

```
-> sflow sampler 1 port 2/1/1-5 receiver 1 rate 2048 sample-hdr-size 128
```

Note. *Optional.* To verify the sFlow sampler configuration, enter **show sflow sampler**, followed by the sFlow sampler instance ID. The display is similar to the one shown below:

```
-> show sflow sampler 1

Instance Interface    Receiver    Rate     Sample-Header-Size
----------------------------------------------------------------
1         2/1/1          1         2048             128
1         2/1/2          1         2048             128
1         2/1/3          1         2048             128
1         2/1/4          1         2048             128
1         2/1/5          1         2048             128
```

For more information about this command, see "sFlow" on page 36-28 or the "sFlow Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide.*

Follow the steps below to create a sFlow poller session.

1 To create a sFlow poller session, use the **sflow poller** command by entering **sflow poller**, followed by the instance ID, port list, receiver, and the interval. For example:

```
-> sflow poller 1 port 1/2/6-10 receiver 1 interval 30
```

Note. *Optional.* To verify the sFlow poller configuration, enter **show sflow poller**, followed by the sFlow poller instance ID. The display is similar to the one shown below:

```
-> show sflow poller
Instance Interface    Receiver  Interval(Secs)
-----------------------------------------
1        2/1/6         1         30
1        2/1/7         1         30
1        2/1/8         1         30
1        2/1/9         1         30
1        2/1/10         1         30
```

For more information about this command, see "sFlow" on page 36-28 or the "sFlow Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide.*

Remote Monitoring (RMON) Overview

The following sections detail the specifications, defaults, and quick set up steps for the RMON feature. Detailed procedures are found in "Remote Monitoring (RMON)" on page 36-33.

RMON Probe Defaults

The following table shows Remote Network Monitoring default values.

Parameter Description	CLI Command	Default Value/Comments
RMON Probe Configuration	**rmon probes**	No RMON probes configured.

Quick Steps for Enabling/Disabling RMON Probes

1 Enable an inactive (or disable an active) RMON probe, where necessary. You can also enable or disable all probes of a particular flavor, if desired. For example:

```
-> rmon probes stats 1011 enable
-> rmon probes history disable
```

2 To verify the RMON probe configuration, enter the **show rmon probes** command, with the keyword for the type of probe. For example, to display the statistics probes, enter the following:

```
-> show rmon probes stats
       Chassis/
Entry  Slot/Port  Flavor    Status      Duration     System Resources
-------+----------+---------+-----------+------------+----------------
   1026 1/1/26     Ethernet  Active      71:49:41     301 bytes
   1025 1/1/25     Ethernet  Active      71:49:20     301 bytes
   1001 1/1/1      Ethernet  Active      71:48:05     300 bytes
```

3 To view statistics for a particular RMON probe, enter the **show rmon probes** command, with the keyword for the type of probe, followed by the entry number for the desired RMON probe. For example:

```
-> show rmon probes 1026
Probe's Owner: Switch Auto Probe on Chassis 1, Slot 1, Port 26, ifindex 1026
    Entry    1026
      Flavor = Ethernet, Status = Active,
      Time = 71 hrs 50 mins,
      System Resources (bytes) = 301
```

For more information about these commands, see "Displaying a List of RMON Probes" on page 36-36, "Displaying Statistics for a Particular RMON Probe" on page 36-37, or the "RMON Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Switch Health Overview

The following sections detail the specifications, defaults, and quick set up steps for the switch health feature. Detailed procedures are found in "Monitoring Switch Health" on page 36-40.

Switch Health Defaults

The following table shows Switch Health default values.

Parameter Description	CLI Command	Default Value/Comments
Resource Threshold Limit Configuration	**health threshold**	80 percent
Sampling Interval Configuration	**health interval**	5 seconds
Switch Temperature	**health threshold**	60 degrees Celsius

Quick Steps for Configuring Switch Health

1 Display the health threshold limits, health sampling interval settings, and/or health statistics for the switch, depending on the parameters you wish to modify. (For best results, note the default settings for future reference.) For example:

```
-> show health configuration
```

The default settings for the command you entered is displayed. For example:

```
Rx Threshold              = 80
TxRx Threshold            = 80
Memory Threshold          = 80
CPU Threshold             = 80
Sampling Interval (Secs)  = 10
```

2 Enter the appropriate command to change the required health threshold or health sampling interval parameter settings or reset all health statistics for the switch. For example:

```
-> health threshold memory 85
```

Note. *Optional.* To verify the Switch Health configuration, enter **show health configuration**, followed by the parameter you modified (like the **memory percent**). The display is similar to the one shown below:

```
Memory Threshold         = 85
```

For more information about this command, see "Displaying Health Threshold Limits" on page 36-42 or the "Health Monitoring Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Port Mirroring

On chassis-based or standalone switches, you can set up port mirroring sessions between Ethernet ports within the same switch.

All Ethernet ports support port mirroring. When port mirroring is enabled, the active “mirrored” port transmits and receives network traffic normally, and the “mirroring” port receives a copy of all transmit and receive traffic to the active port. You can connect an RMON probe or network analysis device to the mirroring port to see an exact duplication of traffic on the mirrored port without disrupting network traffic to and from the mirrored port.

Port mirroring runs in the Chassis Management software and is supported for Ethernet ports. In addition, the switch supports “N-to-1” port mirroring, where up to 128 source ports can be mirrored to a single destination port.

Refer to the Port Mirroring Specifications Table in the “Port Mirroring Overview” on page 36-3 for the number of mirroring sessions supported.

What Ports Can Be Mirrored?

Mirroring between any similar ports and between any SFP to any other SFP port is supported.

How Port Mirroring Works

When a frame is received on a mirrored port, it is copied and sent to the mirroring port. The received frame is actually transmitted twice across the switch backplane–once for normal bridging and then again to the mirroring port.

When a frame is transmitted by the mirrored port, a copy of the frame is made, tagged with the mirroring port as the destination, and sent back over the switch backplane to the mirroring port. The diagram below illustrates the data flow between the mirrored and mirroring ports.

Note. When port mirroring is enabled, there may be some performance degradation, since all frames received and transmitted by the mirrored port need to be copied and sent to the mirroring port.

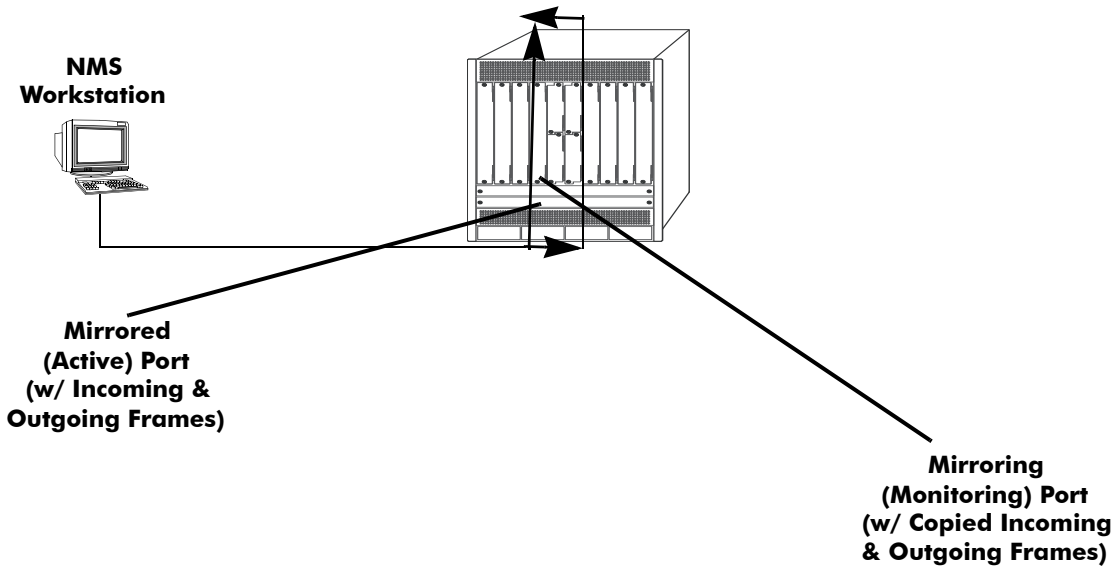


Figure 36-1 : Relationship Between Mirrored and Mirroring Ports

What Happens to the Mirroring Port

Mirroring Port (MTP), can not be assigned to a port with Tagged VLAN configured on it. Once the Mirroring Port (MTP) is configured the port does not belong to any VLAN. Inbound traffic into the MTP is dropped, since it does not belong to any VLAN. When unblocked VLAN is configured, the VLAN ID specified is assigned to the MTP port as the default VLAN. Hence allowing inbound traffic and handling traffic for that VLAN ID. Spanning tree remains disabled on MTP port.

Mirroring on Multiple Ports

If mirroring is enabled on multiple ports and the same traffic is passing through these ports, then only one copy of each packet is sent to the mirroring destination. When the packet is mirrored for the first time, the switching ASIC flags the packet as “already mirrored” If the packet goes through one more port where mirroring is enabled, that packet is not mirrored again. If both mirroring and monitoring are enabled then the packet is either mirrored or monitored (that is sent to CPU), whichever comes first.

Using Port Mirroring with External RMON Probes

Port mirroring is a helpful monitoring tool when used in conjunction with an external RMON probe. Once you set up port mirroring, the probe can collect all relevant RMON statistics for traffic on the mirrored port. You can also move the mirrored port so that the mirroring port receives data from different ports. In this way, you can roam the switch and monitor traffic at various ports.

Note. If the mirroring port monitors mirrored traffic on an RMON probe belonging to a different VLAN than the mirrored port, it must be protected from blocking due to Spanning Tree updates. See “Unblocking Ports (Protection from Spanning Tree)” on page 36-15 for details.

The diagram on the following page illustrates how port mirroring can be used with an external RMON probe to copy RMON probe frames and Management frames to and from the mirroring and mirrored ports. Frames received from an RMON probe attached to the mirroring port can be seen as being received by the mirrored port. These frames from the mirroring port are marked as if they are received on the

mirrored port before being sent over the switch backplane to an NMS station. Therefore, management frames destined for the RMON probe are first forwarded out of the mirrored port. After being received on the mirrored port, copies of the frames are mirrored out of the mirroring port—the probe attached to the mirroring port receives the management frames.

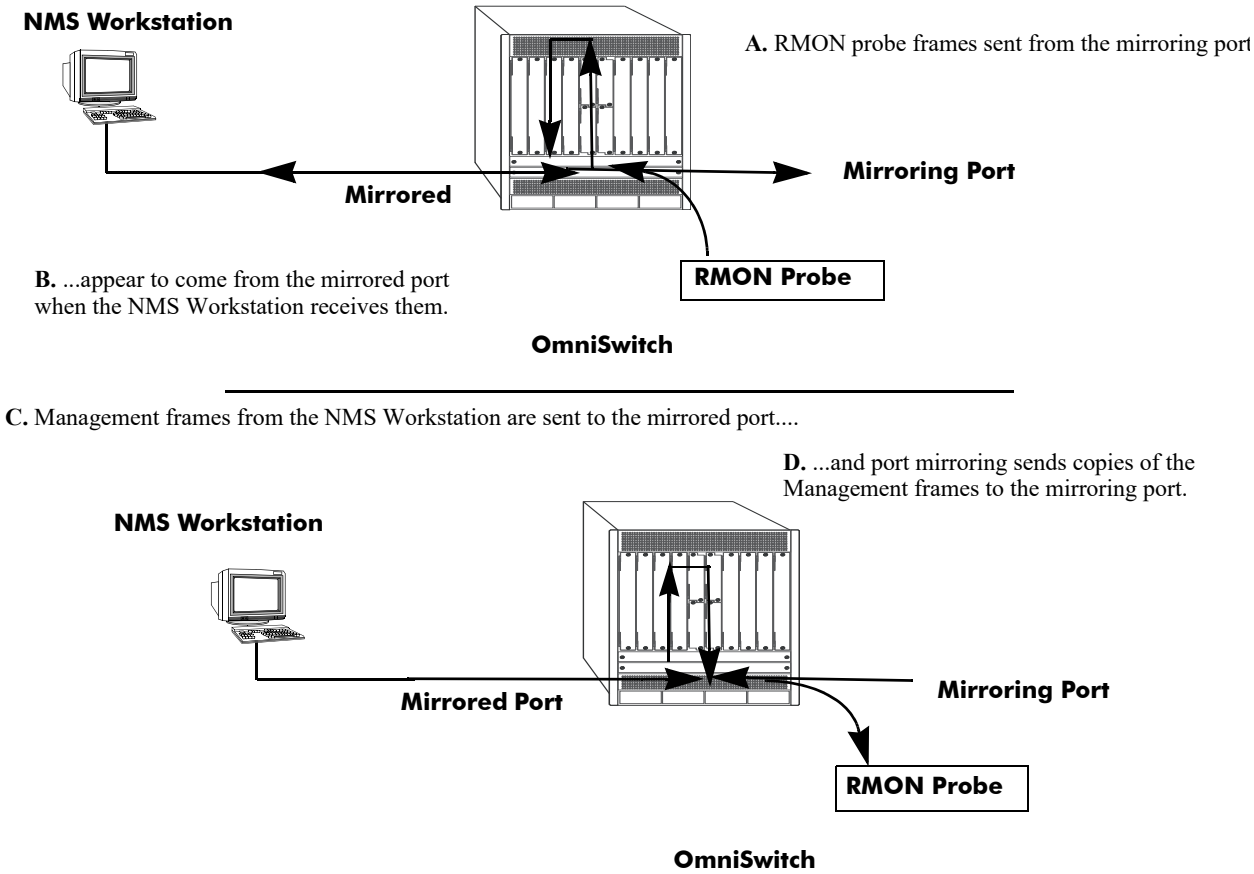


Figure 36-2 : Port Mirroring Using External RMON Probe

Remote Port Mirroring

Remote Port Mirroring expands the port mirroring functionality by allowing mirrored traffic to be carried over the network to a remote switch. With Remote Port Mirroring the traffic is carried over the network using a dedicated Remote Port Mirroring VLAN, no other traffic is allowed on this VLAN. The mirrored traffic from the source switch is tagged with the VLAN ID of the Remote Port Mirroring VLAN and forwarded over the intermediate switch ports to the destination switch where an analyzer is attached.

Since Remote Port Mirroring requires traffic to be carried over the network, the following exceptions to regular port mirroring exist:

- Spanning Tree must be disabled for the Remote Port Mirroring VLAN on all switches.
- There must not be any physical loop present in the Remote Port Mirroring VLAN.
- Remote port mirroring (RPMIR) MTP port can have tagged VLAN and untagged default VLAN on it.
- The VLAN ID used for RPMIR cannot be assigned to the MTP port.
- The VLAN ID used for RPMIR cannot be assigned to the unblocked VLAN.
- On the intermediate and destination switches, source learning must be disabled or overridden on the ports belonging to the Remote Port Mirroring VLAN.
- The **mac-learning vlan disable** command can be used to override source learning on an OmniSwitch.

The following types of traffic are not mirrored:

- Link Aggregation Control Packets (LACP)
- 802.1AB (LLDP)
- 802.1x port authentication
- 802.3ag (OAM)
- Layer 3 control packets
- Generic Attribute Registration Protocol (GARP)

For more information and an example of a Remote Port Mirroring configuration, see “Remote Port Mirroring” on page 36-13.

Creating a Mirroring Session

Before port mirroring can be used, it is necessary to create a port mirroring session. The **port-mirroring source destination** CLI command can be used to create a mirroring session between a mirrored (active) port and a mirroring port. Two (2) port mirroring sessions are supported in a standalone switch. In addition, "N-to-1" port mirroring is supported, where up to 128 source ports can be mirrored to a single destination port.

To create a mirroring session, enter the **port-mirroring source destination** command and include the port mirroring session ID number and the source and destination chassis/slot/ports, as shown in the following example:

```
-> port-mirroring 6 source port 1/2/3 destination port 1/2/4
```

This command line specifies mirroring session 6, with the source (mirrored) port located in chassis 1/slot 2/port 3, and the destination (mirroring) port located in chassis 1/slot 3/port 4.

To create a remote port mirroring session, enter the **port-mirroring source destination** command and include the port mirroring session ID number, the source and destination chassis/slot/ports, and the remote port mirroring VLAN ID as shown in the following example:

```
-> port-mirroring 8 source port 1/1/1 destination port 1/1/2 rpmir-vlan 1000
```

This command line specifies remote port mirroring session 8, with the source (mirrored) port located on slot 1/port 1, the destination (mirroring) port on chassis 1/slot 1/port 2, and the remote port mirroring VLAN 1000.

Creating an "N-to-1" port mirroring session is supported, where up to 128 source ports can be mirrored to a single destination port. In the following example, port 1/2, 2/1, and 2/3 are mirrored on destination port 1/2/4 in session 1:

```
-> port-mirroring 1 source port 1/1/2 destination port 1/2/4
-> port-mirroring 1 source port 1/2/1 destination port 1/2/4
-> port-mirroring 1 source port 1/2/3 destination port 1/2/4
```

As an option, you can specify a range of source ports and/or multiple source ports. In the following example, ports 1/2 through 1/6 are mirrored on destination port 1/2/4 in session 1:

```
-> port-mirroring 1 source port 1/1/2-6 destination port 1/2/4
```

In the following example, ports 1/9, 2/7, and 3/5 are mirrored on destination port 1/2/4 in session 1:

```
-> port-mirroring 1 source port 1/1/9 1/2/7 1/3/5 destination port 1/2/4
```

In the following example, 1/2 through 1/6 and 1/9, 2/7, and 3/5 are mirrored on destination port 1/2/4 in session 1:

```
-> port-mirroring 1 source port 1/1/2-6 1/1/9 1/2/7 1/3/5 destination port 1/2/4
```

Note. Ports can be added after a port mirroring session has been configured.

Policy Based Multiple Destination Mirroring

Policy based multiple destination mirroring ports and link aggregates is supported. For more information and an example of a policy based multiple destination mirroring configuration, see "Configuring Policy Based Multiple Destination Mirroring" on page 36-19

Unblocking Ports (Protection from Spanning Tree)

Spanning tree is disabled by default on an MTP port. When unblocked VLAN is configured , the VLAN ID specified is assigned to the MTP port as the default VLAN. Hence allowing inbound traffic and handling traffic for that VLAN ID. Spanning tree remains disabled. To create a mirroring session that protects the mirroring port from being blocked *(default),* enter the **port-mirroring source destination** CLI command and include the port mirroring session ID number, source and destination slot/ports, and unblocked VLAN ID number, as shown in the following example:

```
-> port-mirroring 6 source port 1/2/3 destination port 1/2/4 unblocked-vlan 750
```

This command line specifies mirroring session 6, with the source (mirrored) port located in chassis 1/slot 2/port 3, and the destination (mirroring) port located in chassis 1/slot 2/port 4. The mirroring port on VLAN 750 is protected from Spanning Tree updates.

Enabling or Disabling Mirroring Status

Mirroring Status is the parameter using which you can enable or disable a mirroring session (i.e., turn port mirroring on or off). There are two ways to do this:

- *Creating a Mirroring Session and Enabling Mirroring Status* or *Disabling a Mirroring Session (Disabling Mirroring Status)*. These procedures are described below and on the following page.
- *Enabling or Disabling a Port Mirroring Session*—"shorthand" versions of the above commands that require fewer keystrokes. Only the port mirroring session ID number needs to be specified, rather than the entire original command line syntax (e.g., source and destination slot/ports and optional unblocked VLAN ID number). See "Enabling or Disabling a Port Mirroring Session (Shorthand)" on page 36-17 for details.

Disabling a Mirroring Session (Disabling Mirroring Status)

To disable the mirroring status of the configured session between a mirrored port and a mirroring port (turning port mirroring off), use the **port-mirroring source destination** CLI command. Be sure to include the port mirroring session ID number and the keyword **disable**.

In this example, the command specifies port mirroring session 6, with the mirrored (active) port located in chassis 1/slot 2/port 3, and the mirroring port located in slot 6/port 4. The mirroring status is disabled (i.e., port mirroring is turned off):

```
-> port-mirroring 6 source port 1/1/1 disable
```

Note. You can modify the parameters of a port mirroring session that has been disabled.

Keep in mind that the port mirroring session configuration remains valid, even though port mirroring has been turned off.

Note. The port mirroring session identifier and slot/port locations of the designated interfaces must always be specified.

Configuring Port Mirroring Direction

By default, port mirroring sessions are bidirectional. To configure the direction of a port mirroring session between a mirrored port and a mirroring port, use the **port-mirroring source destination** CLI command by entering port mirroring, followed by the port mirroring session ID number, the source and destination slot/ports, and **bidirectional**, **inport**, or **outport**.

Note. Optionally, you can also specify the optional unblocked VLAN ID number and either **enable** or **disable** on the same command line.

In this example, the command specifies port mirroring session 6, with the mirrored (active) port located in chassis 1/slot 2/port 3 and the mirroring port located in slot 6/port 4. The mirroring direction is unidirectional and inward bound:

```
-> port-mirroring 6 source port 1/2/3 destination port 1/6/4 inport
```

In this example, the command specifies port mirroring session 6, with the mirrored (active) port located in chassis 1/slot 2/port 3, and the mirroring port located in chassis 1/slot 6/port 4. The mirroring direction is unidirectional and outward bound:

```
-> port-mirroring 6 source port 1/2/3 destination port 1/6/4 outport
```

You can use the bidirectional keyword to restore a mirroring session to its default bidirectional configuration. For example:

```
-> port-mirroring 6 source port 1/2/3 destination port 1/6/4 bidirectional
```

Note. The port mirroring session identifier and slot/port locations of the designated interfaces must always be specified.

Destination Tag-remove

Use this option to remove the VLAN tag on mirrored traffic that egresses out of the destination mirroring ports. Enter the **port-mirroring source destination** CLI command and include the port mirroring session ID number, source and destination chassis/slot/ports or link aggregate as shown in the following example:

```
-> port-mirroring 7 source port 1/2/3 destination linkagg 3 tag-remove
```

Enabling or Disabling a Port Mirroring Session (Shorthand)

Once a port mirroring session configuration has been created, this command is useful for enabling or disabling it (turning port mirroring on or off) without having to re-enter the source and destination ports and unblocked VLAN ID command line parameters.

To enable a port mirroring session, enter the **port-mirroring** command, followed by the port mirroring session ID number and the keyword **enable**. The following command enables port mirroring session 6 (turning port mirroring on):

```
-> port-mirroring 6 enable
```

Note. Port mirroring session parameters cannot be modified when a mirroring session is enabled. Before you can modify parameters, the mirroring session must be disabled.

To disable a port mirroring session, enter the **port-mirroring** command, followed by the port mirroring session ID number and the keyword **disable**. The following command disables port mirroring session 6 (turning port mirroring off):

```
-> port-mirroring 6 disable
```

Displaying Port Mirroring Status

To display port mirroring status, use the **show port-mirroring status** command. To display all port mirroring sessions, enter:

```
   -> show port-mirroring status 6
Session    Mirror        Mirror      Unblocked   RPMIR     Config    Oper
           Destination    Direction     Vlan       Vlan     Status    Status
----------+-----------+----------------+--------+---------+-------+---------
     1.     1/1/11       bidirectional  NONE      NONE      Enable    on
----------+-----------+----------------+--------+---------+---------+---------
         Mirror
         Source
----------+-----------+--------------+----------+---------+----------+---------
     1.      1/1/2       bidirectional     -         -      Enable    On
     1.      1/1/3       bidirectional     -         -      Enable    On
     1.      1/1/4       bidirectional     -         -      Enable    On
     1.      1/1/5       bidirectional     -         -      Enable    On
```

Deleting A Mirroring Session

The **no** form of the **port-mirroring** command can be used to delete a previously created mirroring session configuration between a mirrored port and a mirroring port.

To delete a mirroring session, enter the **no port-mirroring** command, followed by the port mirroring session ID number. For example:

```
-> no port-mirroring 6
```

In this example, port mirroring session 6 is deleted.

Note. The port mirroring session identifier must always be specified.

Configuring Remote Port Mirroring

This section describes the steps required to configure Remote Port Mirroring between Source, Intermediate, and Destination switches.

The following diagram shows an example of a Remote Port Mirroring configuration:

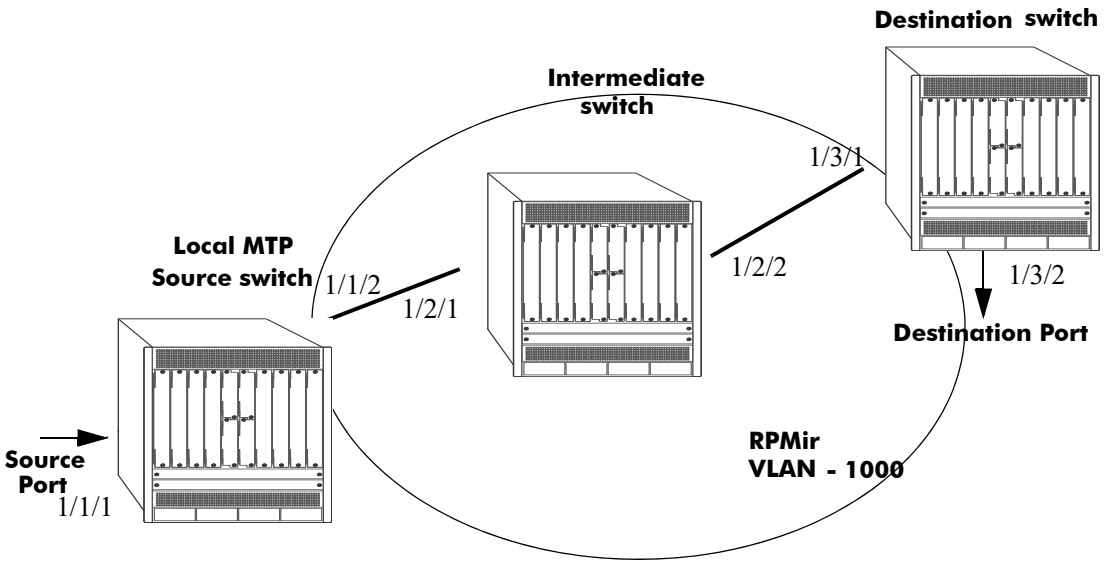


Figure 36-3 : Remote Port Mirroring Example

Configuring Source Switch

Follow the steps given below to configure the Source Switch:

```
-> port-mirroring 8 source port 1/1/1
-> port-mirroring 8 destination port 1/1/2 rpmir-vlan 1000
```

Configuring Intermediate Switch

Follow the steps given below to configure all the Intermediate Switches:

```
-> vlan 1000
-> spantree vlan 1000 admin-state disable
-> vlan 1000 members port 1/2/1-2 tagged
```

Enter the following QoS commands to override source learning:

```
-> policy condition c_is1 source vlan 1000
-> mac-learning vlan 1000 disable
-> policy action a_is1 redirect port 1/2/2
-> policy rule r_is1 condition c_is1 action a_is1
-> qos apply
```

Note. If the intermediate switches are not OmniSwitches, refer to the vendor documentation for instructions on disabling or overriding source learning.

Configuring Destination Switch

Follow the steps given below to configure the Destination Switch:

```
-> vlan 1000
-> spantree vlan 1000 admin-state disable
-> vlan 1000 members port 1/3/1-2 tagged
```

Enter the following QoS commands to override source learning:

```
-> policy condition c_ds1 source vlan 1000
-> mac-learning vlan 1000 disable
-> policy action a_ds1 redirect port 1/3/2
-> policy rule r_ds1 condition c_ds1 action a_ds1
-> qos apply
```

Configuring Policy Based Multiple Destination Mirroring

This section describes the steps required to configure policy based multiple destination mirroring. The following diagram shows an example of a policy based multiple destination mirroring configuration:

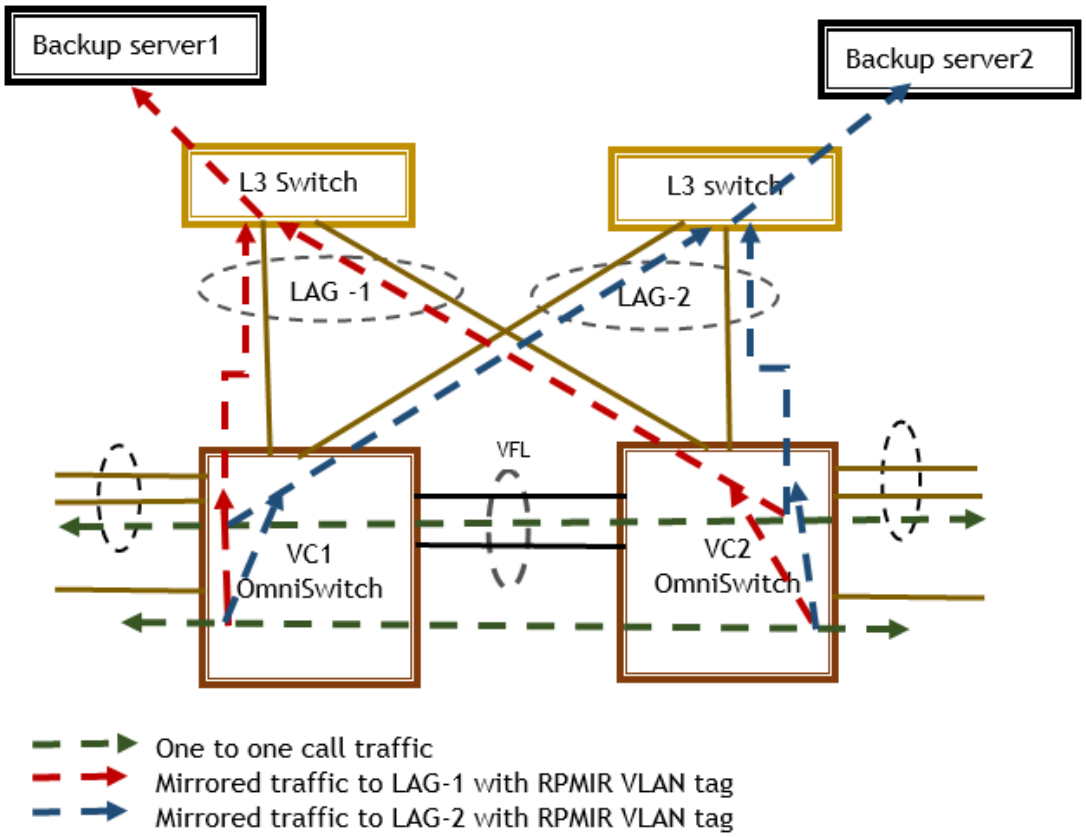


Figure 36-4 : Policy Based Multiple Destination Mirroring

To mirror the traffic of specific user MACs to backup servers, following configurations are required:

Configuration on OmniSwitch

Configure RPMIR VLAN on port mirroring session to send the mirrored traffic on specific VLAN domain, which is used to reach backup servers.

```
-> port-mirroring 1 destination linkagg 10 rpmir-vlan 100 enable
-> port-mirroring 1 destination linkagg 20 rpmir-vlan 100 enable
-> port-mirroring 1 enable
```

Apply QoS policy on specific user MACs and mirror the traffic to the mirror session.

```
-> policy condition c1 source mac 00:00:00:11:22:33 mask FF:FF:FF:00:00:00
-> policy action a1 mirror session 1   (Port mirroring session 1)
-> policy rule r1 condition c1 action a1 precedence 100
-> qos apply
```

Configuration on Adjacent L3 Switches

Configure RPMIR VLAN, which is used to forward the mirrored traffic to backup servers. MAC learning must be disabled on the RPMIR VLAN.

```
-> vlan 100 admin-state enable
-> vlan 100 members port 1/3/4 tagged (port connecting to backup server)
-> mac-learning vlan 100 disable
```

Configuring Remote Port Mirroring (RPMIR) Loopback Session for Multiple Destination Ports

RPMIR Loopback session sends the mirrored traffic from the source port to multiple destination ports associated with the source port.

When the Loopback mode is enabled, the mirrored traffic sent to the Loopback port of RPMIR will be looped and sent to the same port as ingress packets. The ingress packets are Q-tagged with the RPMIR VLANID. The destination ports that are associated with RPMIR VLANID egresses out the mirrored RPMIR traffic, so that ingress mirrored packets are allowed.

Enabling RPMIR Loopback Session

The Loopback mode can be configured only with the RPMIR session. To enable the Loopback mode for the RPMIR session, following exceptions must be considered:

- The destination port associated to a RPMIR session must not be Q-tagged with any other VLANID other than the RPMIR VLAN.
- The source port must not be Q-tagged using the same VLANID which is associated to the destination port of the RPMIR session on which the Loopback is enabled.
- No other traffic must be allowed through the RPMIR VLAN.
- The STP state of Loopback port will be changed to "blocking" when PMM session is running on RPMIR-VLAN.
- STP must be enabled on the VLANID which is tagged as default on the destination port of RPMIR-Loopback session to prevent loop on destination ports.

Use the **port-mirroring source destination** CLI to enable the Loopback mechanism for RPMIR session. For example:

```
-> port-mirroring 1 source port 1/1/1 destination port 1/1/3 bidirectional
rpmir-vlan 100 loopback enable
```

Note. Source learning on Loopback port will be disabled for RPMIR session when Loopback mechanism is enabled.

Disabling RPMIR Loopback Session

To disable the Loopback for RPMIR session, the entire session must be removed and reconfigured without the Loopback mode.

Use the **port-mirroring source destination** CLI to disable the Loopback mechanism for RPMIR session. For example:

```
-> no port mirroring 1
-> port mirroring 1 source port 1/1/1 destination port 1/1/3 bidirectional
rpmir-vlan 100 enable
```

To verify the RPMIR Loopback status of the destination port, use the **show port mirroring status** CLI.

Sample RPMIR Loopback Session Configuration

The following diagram shows an example of RPMIR Loopback session configuration.

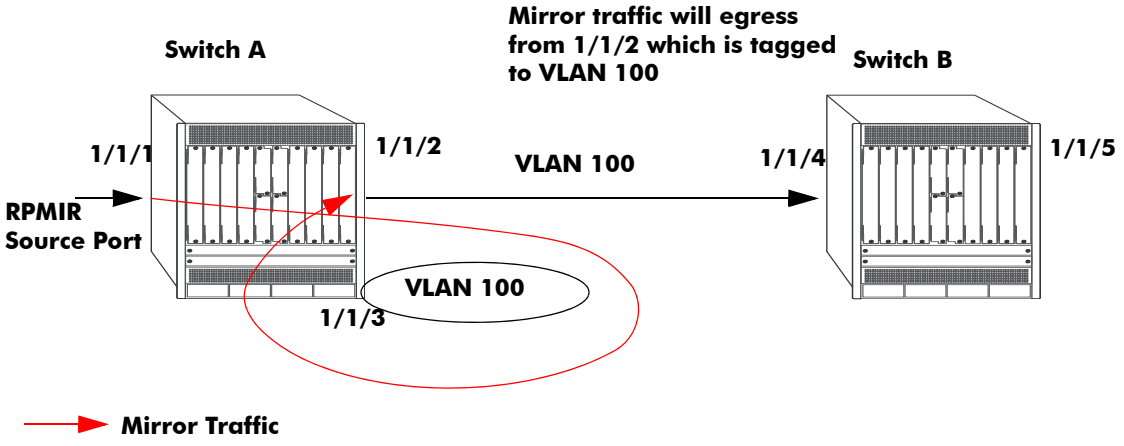


Figure 36-5 : RPMIR Loopback Session Configuration

In the sample configuration, traffic from the port 1/1/1 is mirrored to port 1/1/3. The port 1/1/3 is configured as the RPMIR loopback port. The mirrored traffic is looped back in port 1/1/3 and egressed out to the ports tagged to RPMIR VLAN 100.

Configuring Source Switch A

Follow the steps given below to configure the switch A:

```
-> vlan 100
```

```
-> vlan 100 members port 1/1/2 tagged
-> vlan 100 members port 1/1/3 tagged
-> port mirroring 1 source port 1/1/1 destination port 1/1/3 bidirectional
rpmir-vlan 100 loopback enable
```

The RPMIR traffic is flooded through all the ports of the source switch which is tagged to RPMIR VLAN 100.

Configuring Destination Switch B

Follow the steps given below to configure the switch B:

```
-> vlan 100
-> vlan 100 stp disable
-> vlan 100 members port 1/1/4 tagged
-> vlan 100 members port 1/1/5 tagged
```

Enter the following QoS commands to override source learning:

```
-> policy condition c_is1 source vlan 100
-> policy action a_is1 redirect port 1/1/3
-> qos apply
```

Setting Port Mirroring Traffic Priority

To set the priority for mirroring traffic to either normal or high, enter the **port-mirroring traffic-priority** command. By default, traffic priority is set to normal. When the traffic priority is set to high, ingress and egress mirroring receive strict priority over other clients (regular received, not mirrored, traffic, and so on). For example:

```
-> port-mirroring traffic-priority high

-> port-mirroring traffic-priority normal
```

Port Monitoring

An essential tool of the network engineer is a network packet capture device. A packet capture device is usually a PC-based computer, such as the Sniffer®, that provides a means for understanding and measuring data traffic of a network. Understanding data flow in a VLAN-based switch presents unique challenges, primarily because traffic moves inside the switch, especially on dedicated devices.

The port monitoring feature allows you to examine packets to and from a specific Ethernet port. Port monitoring has the following features:

- Software commands to enable and display captured port data.
- Captures data in Network General® file format.
- A file called **pmonitor.enc** is created in the **/flash** memory when you configure and enable a port monitoring session.
- Data packets time stamped.
- One port monitored at a time.
- RAM-based file system.
- Statistics gathering and display.

The port monitoring feature also has the following restrictions:

- All packets cannot be captured. (Estimated packet capture rate is around 500 packets/second.)
- The maximum number of monitoring sessions is limited to one per chassis.
- Only the first 64 bytes of the traffic is captured in 'brief' mode. If the monitoring capture-type is set to 'full' the entire packet is captured.
- Link Aggregation ports can be monitored.
- If both mirroring and monitoring are enabled, then packets is either mirrored *or* monitored (i.e., sent to CPU), whichever comes first. See "Mirroring on Multiple Ports" on page 36-11 for more information.

You can select to dump real-time packets to a file. Once a file is captured, you can FTP it to a Sniffer or PC for viewing.

Configuring a Port Monitoring Session

To configure a port monitoring session, use the **port-monitoring source** command by entering **port -monitoring**, followed by the user-specified session ID number, **source**, the slot number of the port to be monitored, a slash (/), and the port number of the port.

For example, to configure port monitoring session 6 on port 1/2/3 enter:

```
-> port-monitoring 6 source port 1/2/3
```

Note. One port monitoring session can be configured per chassis.

In addition, you can also specify optional parameters shown in the table below. These parameters must be entered after the slot and port number.

keywords		
file	**no file**	**size**
no overwrite	**inport**	**outport**
bidirectional	**timeout**	**enable**
disable	**capture-type**	**full**
brief		

For example, to configure port monitoring session 6 on port 1/2/3 and administratively enable it, enter:

```
-> port-monitoring 6 source port 1/2/3 enable
```

These keywords can be used when creating the port monitoring session or afterwards. See the sections below for more information on using these keywords.

Enabling a Port Monitoring Session

To disable a port monitoring session, use the **port-monitoring source** command by entering **port - monitoring**, followed by the user-specified session ID number, **source**, the slot number of the port to be monitored, a slash (/), the port number of the port, and **enable**. For example, to enable port monitoring session 6 on port 1/2/3, enter:

```
-> port-monitoring 6 source port 1/2/3 enable
```

Disabling a Port Monitoring Session

To disable a port monitoring session, use the **port-monitoring** command by entering **port-monitoring**, followed by the port monitoring session ID and **pause**. For example, to disable port monitoring session 6, enter:

```
-> port-monitoring 6 disable
```

Deleting a Port Monitoring Session

To delete a port monitoring session, use the **no** form of the **port-monitoring** command by entering **no port-monitoring**, followed by the port monitoring session ID. For example, to delete port monitoring session 6, enter:

```
-> no port-monitoring 6
```

Pausing a Port Monitoring Session

To pause a port monitoring session, use the **port-monitoring** command by entering **port-monitoring**, followed by the port monitoring session ID and **pause**. For example, to pause port monitoring session 6, enter:

```
-> port-monitoring 6 pause
```

To resume a paused port monitoring session, use the **port-monitoring** command by entering **port -monitoring**, followed by the port monitoring session ID and **resume**. For example, to resume port monitoring session 6, enter:

```
-> port-monitoring 6 resume
```

Configuring Port Monitoring Session Persistence

By default, a port monitoring session is enabled. To modify the length of time before a port monitoring session is disabled from 0 (the default, where the session is permanent) to 2147483647 seconds, use the **port-monitoring source** CLI command by entering **port-monitoring**, followed by the user-specified session ID number, **source**, the slot number of the port to be monitored, a slash (/), the port number of the port, **timeout**, and the number of seconds before it is disabled.

For example, to configure port monitoring session 6 on port 1/2/3 that lasts 12000 seconds before it is disabled, enter:

```
-> port-monitoring 6 source port 1/2/3 timeout 12000
```

Configuring a Port Monitoring Data File

By default, a file called **pmonitor.enc** is created in the **/flash** directory when you configure and enable a port monitoring session. This file can be FTPed for later analysis. To configure a user-specified file, use the **port-monitoring source** CLI command by entering **port-monitoring**, followed by the user-specified session ID number, **source**, the slot number of the port to be monitored, a slash (/), the port number of the port, **file**, and the name of the file. The port monitoring sniffer file can be viewed using software such as wireShark or ethereal.

For example, to configure port monitoring session 6 on port 1/2/3 with a data file called “user_port” in the **/flash** directory, enter:

```
-> port-monitoring 6 source port 1/2/3 file /flash/user_port
```

Optionally, you can also configure the size of the file and/or you can configure the data file so that more recent packets do not overwrite older packets in the data file if the file size is exceeded.

To create a file and configure its size, use the **port-monitoring source** CLI command by entering **port-monitoring**, followed by the user-specified session ID number, **source**, the slot number of the port to be monitored, a slash (/), the port number of the port, **file**, the name of the file, **size**, and the size of the file in 16K byte increments.

For example, to configure port monitoring session 6 on port 1/2/3 with a data file called “user_port” in the **/flash** directory with a size of 49152 (3 * 16K), enter:

```
-> port-monitoring 6 source port 1/2/3 file /flash/user_port size 3
```

To select the type of port monitoring information captured, use the **port-monitoring source** CLI command by entering **port-monitoring**, followed by the user-specified session ID number, **source**, the slot number of the port to be monitored, a slash (/), the port number of the port, **file**, the name of the file, and the **capture-type** keyword followed by the keywords, **full** or **brief**.

For example, to configure port monitoring session 6 on port 1/2/3 with a data file called "user_port" in the **/flash** directory with a size of 49152 (3 * 16K), and port monitoring **capture-type full,** enter:

```
-> port-monitoring 6 source port 1/2/3 file /flash/user_port 3 capture-type full
```

To prevent more recent packets from overwriting older packets in the data file, if the file size is exceeded, use the **port-monitoring source** CLI command by entering **port-monitoring**, followed by the user-specified session ID number, **source**, the slot number of the port to be monitored, a slash (/), the port number of the port, **file**, the name of the file, and **overwrite off**.

For example, to configure port monitoring session 6 on port 1/2/3 with a data file called "user_port" in the **/flash** directory that does not overwrite older packets if the file size is exceeded, enter:

```
-> port-monitoring 6 source port 1/2/3 file user_port overwrite off
```

To allow more recent packets from overwriting older packets in the data file if the file size is exceeded (the default), use the **port-monitoring source** CLI command by entering **port-monitoring**, followed by the user-specified session ID number, **source**, the slot number of the port to be monitored, a slash (/), the port number of the port, **file**, the name of the file, and **overwrite on**.

For example, to configure port monitoring session 6 on port 1/2/3 with a data file called "user_port" in the **/flash** directory that does not overwrite older packets if the file size is exceeded, enter:

```
-> port-monitoring 6 source port 1/2/3 file /flash/user_port overwrite on
```

Note. The **size** and **no overwrite** options can be entered on the same command line.

Configuring Port Monitoring Direction

By default, port monitoring sessions are bidirectional. To configure the direction of a port mirroring session between a mirrored port and a mirroring port, use the **port-monitoring source** CLI command by entering **port-monitoring**, followed by the user-specified session ID number, **source**, the slot number of the port to be monitored, a slash (/), the port number of the port, and **inport**, **outport**, or **bidirectional**.

For example, to configure port monitoring session 6 on port 1/2/3 as unidirectional and inward bound, enter:

```
-> port-monitoring 6 source port 1/2/3 inport
```

To configure port monitoring session 6 on port 1/2/3 as unidirectional and outward bound, for example, enter:

```
-> port-monitoring 6 source port 1/2/3 outport
```

For example, to restore port monitoring session 6 on port 1/2/3 to its bidirectional direction, enter:

```
-> port-monitoring 6 source port 1/2/3 bidirectional
```

Configuring the Capture Type

To configure the amount of data to be captured, use the **port-monitoring source capture-type** command. If the capture type mode is set to 'brief', only the first 64 bytes of packets will be captured. If the capture type mode is set to 'full', then the full packet is captured regardless of the packet size.

For example, to configure port monitoring session 6 on port 1/2/3 to capture only the first 64 bytes of the packet, enter

```
-> port-monitoring 6 source port 1/2/3 capture-type brief
```

To configure port monitoring session 6 on port 1/2/3 to capture full packet, enter

```
-> port-monitoring 6 source port 1/2/3 capture-type full
```

Displaying Port Monitoring Status and Data

A summary of the show commands used for displaying port monitoring status and port monitoring data is given here:

show port-monitoring status	Displays port monitoring status.
show port-monitoring file	Displays port monitoring data.

For example, to display port monitoring data, use the **show port-monitoring file** command as shown below:

```
-> show port-monitoring file

Destination       | Source            |  Type   | Data
-------------------------------------------------------------------------------
01:80:C2:00:00:00 | 00:20:DA:8F:92:C6 |  BPDU   | 00:26:42:42:03:00:00:00:00:00
00:20:DA:C7:2D:D6 | 08:00:20:95:F3:89 |  UDP    | 08:00:45:00:00:6B:FE:4A:40:00
00:20:DA:A3:89:F6 | 08:00:20:95:F3:89 |  UDP    | 08:00:45:00:00:6B:CF:89:40:00
00:20:DA:BF:5B:76 | 08:00:20:95:F3:89 |  UDP    | 08:00:45:00:00:6B:CF:85:40:00
00:20:DA:A3:89:F6 | 08:00:20:95:F3:89 |  UDP    | 08:00:45:00:00:6B:CF:8A:40:00
00:20:DA:BF:5B:76 | 08:00:20:95:F3:89 |  UDP    | 08:00:45:00:00:6B:CF:86:40:00
00:20:DA:A3:89:F6 | 08:00:20:95:F3:89 |  UDP    | 08:00:45:00:00:6B:CF:8B:40:00
01:80:C2:00:00:00 | 00:20:DA:8F:92:C6 |  BPDU   | 00:26:42:42:03:00:00:00:00:00
00:20:DA:BF:5B:76 | 08:00:20:95:F3:89 |  UDP    | 08:00:45:00:00:6B:CF:87:40:00
```

Note. For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. The **show port-monitoring** command displays only 170 packets from the port monitor file.

sFlow

sFlow is a network monitoring technology that gives visibility in to the activity of the network, by providing network usage information. It provides the data required to effectively control and manage the network usage. sFlow is a sampling technology that meets the requirements for a network traffic monitoring solution.

sFlow is an industry standard with many vendors delivering products with this support. Some of the applications of the sFlow data include:

- Detecting, diagnosing, and fixing network problems
- Real-time congestion management
- Detecting unauthorized network activity
- Usage accounting and billing
- Understanding application mix
- Route profiling and peer optimization
- Capacity planning

sFlow is a sampling technology embedded within switches/routers. It provides the ability to monitor the traffic flows. It requires a sFlow agent software process running as part of the switch software and a sFlow collector which receives and analyses the monitored data. The sFlow collector makes use of SNMP to communicate with a sFlow agent in order to configure sFlow monitoring on the device (switch).

sFlow agent running on the switch/router, combines interface counters and traffic flow (packet) samples preferably on all the interfaces into sFlow datagrams that are sent across the network to an sFlow collector.

Packet sampling on the switch/router is typically performed by the switching/routing ASICs, providing wire-speed performance. In this case, sFlow agent does very little processing, by packaging data into sFlow datagrams that are immediately sent on network. This minimizes the memory and CPU utilization by sFlow agent.

sFlow Manager

The sFlow manager is the controller for all the modules. It initializes all other modules. It interfaces with the Ethernet driver to get the counter samples periodically and reads sampled packets. The counter samples are given to the poller and sampled packets are given to the sampler to format a UDP packet.

Each sFlow manager instance has multiples of receiver, sampler, and poller instances. Each user programmed port has an individual sampler and poller. The sampler and poller could be potentially pointing to multiple receivers if the user has configured multiple destination hosts.

Receiver

The receiver module has the details about the destination hosts where the sFlow datagrams are sent out. If there are multiple destinations then each destination has an instance of the receiver. All these receivers are attached to the sFlow manager instance and to an associated sample/poller.

Sampler

The sampler is the module which gathers samples and fills up the sampler part of the UDP datagram.

Poller

The poller is the module which gets counter samples from Ethernet driver and fills up the counter part of the UDP datagram.

Configuring a sFlow Session

To configure a sFlow receiver session, use the **sflow receiver** command by entering **sflow receiver**, followed by the receiver_index, name, the name of the session and **address**, and the IP address of the receiver.

For example, to configure receiver session 6 on switch 10.255.11.28, enter:

```
-> sflow receiver 6 name sflowtrend address 10.255.11.28
```

In addition, you can also specify optional parameters shown in the table below. These parameters can be entered after the IP address.

keywords	
timeout	**packet-size**
forever	**version**
udp-port	

For example, to configure sFlow receiver session 6 on switch 10.255.11.28 and to specify the packet-size and timeout value, enter:

```
-> sflow receiver 6 name sflowtrend address 10.255.11.28 packet-size 1400
timeout 600
```

To configure a sFlow sampler session, use the **sflow sampler** command by entering **sflow sampler**, followed by the instance ID number, the slot number of the port to be monitored, a slash (/), and the port number and **receiver**, the receiver_index.

For example, to configure sampler session 1 on port 1/2/3, enter:

```
-> sflow sampler 1 port 1/2/3 receiver 6
```

In addition, you can also specify optional parameters shown in the table below. These parameters can be entered after the receiver index.

keywords
rate
sample-hdr-size

For example, to configure sFlow sampler session 1 on port 1/2/3 and to specify the rate and sample-hdr-size, enter:

```
-> sflow sampler 1 port 1/2/3 receiver 6 rate 512 sample-hdr-size 128
```

To configure a sFlow poller session, use the **sflow poller** command by entering **sflow poller**, followed by the instance ID number, the slot number of the port to be monitored, a slash (/), and the port number of the port and **receiver**, then *receiver_index*.

For example, to configure poller session 3 on port 1/1/1, enter:

```
-> sflow poller 3 port 1/1/1 receiver 6
```

In addition, you can also specify the optional **interval** parameter after the receiver index value. For example, to configure sFlow poller session 3 on port 1/1/1 with an interval of 5, enter:

```
-> sflow poller 3 port 1/1/1 receiver 6 interval 5
```

Configuring a Fixed Primary Address

In order to generate the IP packtes and sent the sFlow data-grams out into the network, sFlow agent requires an IP address configured to it. The agents IP address can be configured using the **sflow agent** command. If there is no IP address configured, then the sFlow data-grams will not be sent to the receiver.

For example, to configure the agent IP address, enter:

```
-> sflow agent ip 198.206.181.3
```

Displaying a sFlow Receiver

The **show sflow receiver** command is used to display the receiver table.

For example, to view the sFlow receiver table, enter the **show sflow receiver** command without specifying any additional parameters. A screen similar to the following example is displayed, as shown below:

```
-> show sflow receiver

Receiver 1
Name       = Golden
Address    = IP_V4 198.206.181.3
UDP Port   = 6343
Timeout    = 65535
Packet Size= 1400
DatagramVer= 5
```

Note. For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Displaying a sFlow Sampler

The **show sflow sampler** command is used to display the sampler table.

For example, to view the sFlow sampler table, enter the **show sflow sampler** command without specifying any additional parameters. A screen similar to the following example is displayed, as shown below:

```
-> show sflow sampler

Instance Interface    Receiver    Rate     Sample-Header-Size
----------------------------------------------------------------
1         2/1/1          1         2048             128
1         2/1/2          1         2048             128
1         2/1/3          1         2048             128
1         2/1/4          1         2048             128
1         2/1/5          1         2048             128
```

Note. For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Displaying a sFlow Poller

The **show sflow poller** command is used to display the poller table.

For example, to view the sFlow poller table, enter the **show sflow poller** command without specifying any additional parameters. A screen similar to the following example is displayed, as shown below:

```
-> show sflow poller

Instance  Interface  Receiver   Interval(Secs)
--------+----------+----------+---------------
   1       1/1/1        1         10
   1       1/1/2        1         30
```

Note. For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Displaying a sFlow Agent

The **show sflow agent** command is used to display the receiver table.

For example, to view the sFlow agent table, enter the **show sflow agent** command without specifying any additional parameters. A screen similar to the following example is displayed, as shown below:

```
-> show sflow agent

Agent Version  = 1.3; Alcatel-Lucent; 6.1.1
Agent IP       = 127.0.0.1
```

Note. For more information about the displays that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Deleting a sFlow Session

To delete a sFlow receiver session, use the release form at the end of the **sflow agent** command by entering **sflow receiver**, followed by the receiver index and **release**. For example, to delete sFlow receiver session 6, enter:

```
-> sflow receiver 6 release
```

To delete a sFlow sampler session, use the no form of the **sflow sampler** command by entering **no sflow sampler**, followed by the instance ID number, the slot number of the port to delete, a slash (/), and the port number of the port, enter:

```
-> no sflow sampler 1 port 1/2/3
```

To delete a sFlow poller session, use the no form of the **sflow poller** command by entering **no sflow poller**, followed by the instance ID number, the slot number of the port to delete, a slash (/), and the port number of the port, enter:

```
-> no sflow poller 3 port 1/1/1
```

Remote Monitoring (RMON)

Remote Network Monitoring (RMON) is an SNMP protocol used to manage networks remotely. *RMON probes* can be used to collect, interpret, and forward statistical data about network traffic from designated active ports in a LAN segment to an NMS (Network Management System) application for monitoring and analysis without negatively impacting network performance. RMON software is fully integrated in the Chassis Management software and works with the Ethernet software to acquire statistical information. However, it does not monitor the CMM module's onboard Ethernet Management port on OmniSwitch chassis-based switches (which is reserved for management purposes).

The following diagram illustrates how an External RMON probe can be used with port mirroring to copy RMON probe frames and Management frames to and from the mirroring and mirrored ports. Frames received from an RMON probe attached to the mirroring port can be seen as being received by the mirrored port. These frames from the mirroring port are marked as if they are received on the mirrored port before being sent over the switch backplane to an NMS station. Therefore, management frames that are destined for the RMON probe are first forwarded out of the mirrored port. After being received on the mirrored port, copies of the frames are mirrored out of the mirroring port—the probe attached to the mirroring port receives the management frames.

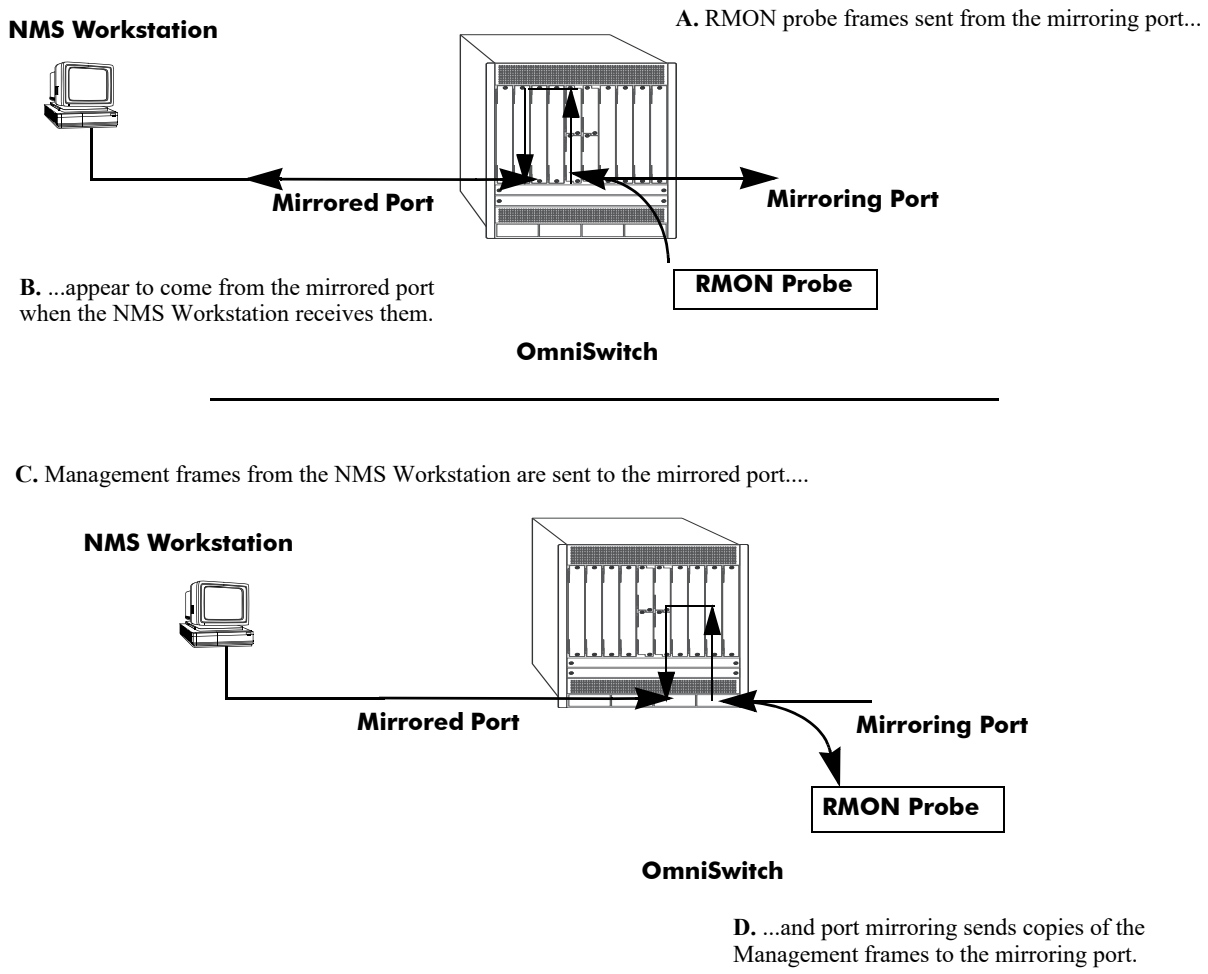


Figure 36-6 : Port Mirroring Using External RMON Probe

RMON probes can be enabled or disabled through CLI commands. Configuration of Alarm threshold values for RMON traps is a function reserved for RMON-monitoring NMS stations.

This feature supports basic RMON 4 group implementation in compliance with RFC 2819, including the **Ethernet Statistics**, **History** (Control & Statistics), **Alarms** and **Events** groups *(described below)*.

Note. RMON 10 group and RMON2 are not implemented in the current release. An external RMON probe that includes RMON 10 group and RMON2 be used where full RMON probe functionality is required.

Ethernet Statistics

Ethernet statistics probes are created whenever new ports are inserted and activated in the chassis. When a port is removed from the chassis or deactivated, the Ethernet statistics group entry associated with the physical port is invalidated and the probe is deleted.

The Ethernet statistics group includes port utilization and error statistics measured by the RMON probe for each monitored Ethernet interface on the switch. Examples of these statistics include CRC (Cyclic Redundancy Check)/alignment, undersized/oversized packets, fragments, broadcast/multicast/unicast, and bandwidth utilization statistics.

History (Control & Statistics)

The History (Control & Statistics) group controls and stores periodic statistical samplings of data from various types of networks. Examples include Utilization, Error Count, and Frame Count statistics.

Alarm

The Alarm group collects periodic statistical samples from variables in the probe and compares them to previously configured thresholds. If a sample crosses a previously configured threshold value, an Event is generated. Examples include Absolute or Relative Values, Rising or Falling Thresholds on the Utilization Frame Count and CRC Errors.

Event

The Event group controls generation and notification of events from the switch to NMS stations. For example, customized reports based on the type of Alarm can be generated, printed and/or logged.

Note. The following RMON groups are not implemented: **Host**, **HostTopN**, **Matrix**, **Filter,** and **Packet Capture**.

Enabling or Disabling RMON Probes

To enable or disable an individual RMON probe, enter the **rmon probes** CLI command. Be sure to specify the type of probe (**stats**/**history**/**alarm**), followed by the entry number (optional), as shown in the following examples.

The following command enables RMON Ethernet Statistics probe number 4012:

```
-> rmon probes stats 4012 enable
```

The following command disables RMON History probe number 10240:

```
-> rmon probes history 10240 disable
```

The following command enables RMON Alarm probe number 11235:

```
-> rmon probes alarm 11235 enable
```

To enable or disable an entire group of RMON probes of a particular flavor type (such as Ethernet Statistics, History, or Alarm), enter the command **without** specifying an *entry-number*, as shown in the following examples.

The following command disables all currently defined (disabled) RMON Ethernet Statistics probes:

```
-> rmon probes stats disable
```

The following command enables all currently defined (disabled) RMON History probes:

```
-> rmon probes history enable
```

The following command enables all currently defined (disabled) RMON Alarm probes:

```
-> rmon probes alarm enable
```

Note. Network activity on subnetworks attached to an RMON probe can be monitored by Network Management Software (NMS) applications.

Displaying RMON Tables

Two separate commands can be used to retrieve and view Remote Monitoring data: **show rmon probes** and **show rmon events**. The retrieved statistics appear in a *table* format (a collection of related data that meets the criteria specified in the command you entered). These RMON tables can display the following kinds of data (depending on the criteria you've specified):

- The **show rmon probes** command can display a list of current RMON probes or statistics for a particular RMON probe.
- The **show rmon events** command can display a list of RMON events (actions that occur in response to Alarm conditions detected by an RMON probe) or statistics for a particular RMON event.

Displaying a List of RMON Probes

To view a list of current RMON probes, enter the **show rmon probes** command with the probe type, without specifying an entry number for a particular probe.

For example, to show a list of the statistics probes, enter:

```
-> show rmon probes stats

        Chassis/
Entry  Slot/Port  Flavor    Status      Duration     System Resources
-------+----------+---------+-----------+------------+----------------
   1026 1/1/26     Ethernet  Active      71:49:41     301 bytes
   1025 1/1/25     Ethernet  Active      71:49:20     301 bytes
   1022 1/1/22     Ethernet  Active      71:48:03     301 bytes
   1023 1/1/23     Ethernet  Active      71:48:03     301 bytes
```

This table entry displays probe statistics for all probes on the switch. The probes are active, utilize 301 bytes of memory, and 71 minutes have elapsed since the last change in status occurred.

To show a list of the history probes, enter:

```
-> show rmon probes history

       Chassis/
Entry  Slot/Port  Flavor    Status      Duration     System Resources
-------+----------+---------+-----------+------------+----------------
     1 1/1/26     History   Active      71:50:08     5471 bytes
     2 1/1/25     History   Active      71:49:47     5471 bytes
     3 1/1/1      History   Active      71:48:32     5470 bytes
     4 1/1/22     History   Active      71:48:30     5471 bytes
     5 1/1/23     History   Active      71:48:30     5471 bytes
```

The table entry displays statistics for RMON History probes on the switch.

To show a list of the alarm probes, enter:

```
-> show rmon probes alarm
       Chassis/
Entry  Slot/Port   Flavor    Status      Duration     System Resources
-------+----------+---------+-----------+------------+----------------
31927    1/1/35    Alarm      Active      00:25:51       608 bytes
```

Displaying Statistics for a Particular RMON Probe

To view statistics for a particular current RMON probe, enter the **show rmon probes** command, specifying an entry number for a particular probe, such as:

```
-> show rmon probes 4005
```

A display showing statistics for the specified RMON probe appears, as shown in the following sections.

Sample Display for Ethernet Statistics Probe

The display shown here identifies RMON Probe 4005's Owner description and interface location (OmniSwitch Auto Probe on chassis 1, slot 4, port 5), Entry number (4005), probe Flavor (Ethernet statistics), and Status (Active). Additionally, the display indicates the amount of time that has elapsed since the last change in status (48 hours, 54 minutes), and the amount of memory allocated to the probe, measured in bytes (301).

```
-> show rmon probes 4005

Probe's Owner: Switch Auto Probe on Chassis 1, Slot 4, Port 5, ifindex 4005
    Entry    4005
      Flavor = Ethernet, Status = Active,
      Time = 48 hrs 54 mins,
      System Resources (bytes) = 301
```

Sample Display for History Probe

The display shown here identifies RMON Probe 10325's Owner description and interface location (chassis 8, slot 1, port 29), the total number of History Control Buckets (samples) requested and granted (50), along with the time interval for each sample (30 seconds) and system-generated Sample Index ID number (287). The probe Entry number identifier (9), probe Flavor (History), and Status (Active), the amount of time that has elapsed since the last change in status (71 hours, 48 minutes), and the amount of memory allocated to the probe, measured in bytes (5471) are also displayed.

```
-> show rmon probes history 30562

Probe's Owner: Switch Auto Probe on Chassis 8, Slot 1, Port 29
  History Control Buckets Requested = 50,
  History Control Buckets Granted   = 50,
  History Control Interval          = 30 seconds,
  History Sample Index              = 287
    Entry       9
      Flavor = History, Status = Active,
      Time = 71 hrs 48 mins,
      System Resources (bytes) = 5471
```

Sample Display for Alarm Probe

The display shown here identifies RMON Probe 11235's Owner description, as well as the Alarm Rising Threshold of the probe and Alarm Falling Threshold, maximum allowable values beyond which an alarm is generated and sent to the Event group (5 and 0, respectively).

Additionally, the corresponding Alarm Rising Event Index number (26020) and Alarm Falling Event Index number (0), which link the Rising Threshold Alarm and Falling Threshold Alarm to events in the Event table, are identified. The Alarm Interval, a time period during which data is sampled (10 seconds) and Alarm Sample Type (delta value—variable) are also shown, as is the Alarm Variable ID number (1.3.6.1.2.1.16.1.1.1.5.4008). The probe Entry number identifier (11235), probe Flavor (Alarm), Status (Active), the amount of time that has elapsed since the last change in status (48 hours, 48 minutes), and the amount of memory allocated to the probe, measured in bytes (1677) are also displayed.

```
-> show rmon probes alarm 31927

Probe's Owner:
   Alarm Rising Threshold      = 5
   Alarm Falling Threshold     = 0
   Alarm Rising Event Index    = 26020
   Alarm Falling Event Index   = 0
   Alarm Interval              = 10 seconds
   Alarm Sample Type           = delta value
   Alarm Startup Alarm         = rising alarm
   Alarm Variable              = 1.3.6.1.2.1.16.1.1.1.5.4008
      Entry 11235
          Flavor = Alarm, Status = Active
          Time = 48 hrs 48 mins,
          System Resources (bytes) = 1677
```

Displaying a List of RMON Events

RMON Events are actions that occur based on Alarm conditions detected by an RMON probe. To view a list of logged RMON Events, enter the **show rmon events** command without specifying an entry number for a particular probe, such as:

```
-> show rmon events
```

A display showing all logged RMON Events must appear, as shown in the following example:

```
Entry      Time             Description
---------+----------------+-------------------------------------------------------
1         00:08:00          etherStatsPkts.4008: [Falling trap] "Falling Event"
2         00:26:00          etherStatsCollisions.2008: "Rising Event"
3         00:39:00          etherStatsCollisions.2008: "Rising Event"
```

The display shown above identifies the Entry number of the specified Event, along with the elapsed time since the last change in status (measured in hours/minutes/seconds) and a description of the Alarm condition detected by the probe for all RMON Logged Events. For example, Entry number 3 is linked to etherStatsCollisions.2008: [Rising trap] "Rising Event," an Alarm condition detected by the RMON probe in which a trap was generated based on a Rising Threshold Alarm, with an elapsed time of 39 minutes since the last change in status.

Displaying a Specific RMON Event

To view information for a specific logged RMON Event, enter the **show rmon events** command, specifying an entry number (event number) for a particular probe, such as:

```
-> show rmon events 3
```

A display showing the specific logged RMON Event must appear, as shown in the following example:

```
Entry      Time            Description
---------+----------------+-------------------------------------------------------
3         00:39:00          etherStatsCollisions.2008: "Rising Event"
```

The display shown above identifies the Entry number of the specified Event, along with the elapsed time since the last change in status (measured in hours/minutes/seconds) and a description of the Alarm condition detected by the probe for the specific RMON Logged Event. For example, Entry number 3 is linked to etherStatsCollisions.2008: [Rising trap] "Rising Event," an Alarm condition detected by the RMON probe in which a trap was generated based on a Rising Threshold Alarm, with an elapsed time of 39 minutes since the last change in status.

Monitoring Switch Health

To monitor resource availability, the NMS (Network Management System) needs to collect significant amounts of data from each switch. As the number of ports per switch (and the number of switches) increases, the volume of data can become overwhelming. The Health Monitoring feature can identify and monitor a switch's resource utilization levels and thresholds, improving efficiency in data collection.

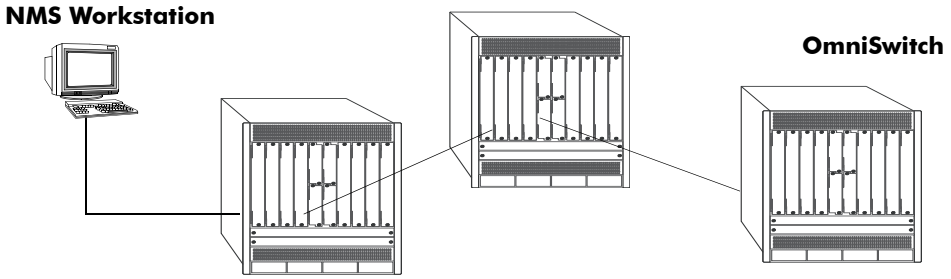


Figure 36-7 : Monitoring Resource Availability from Multiple Ports and Switches

Health Monitoring provides the following data to the NMS:

- Switch-level Input/Output, Memory and CPU Utilization Levels
- Module-level and Port-level Input/Output Utilization Levels

For each monitored resource, the following variables are defined:

- Most recent utilization level (percentage)
- Average utilization level over the last minute (percentage)
- Average utilization level over the last hour (percentage)
- Maximum utilization level over the last hour (percentage)
- Threshold level

Additionally, Health Monitoring provides the capacity to specify thresholds for the resource utilization levels it monitors and generates traps based on the specified threshold criteria.

The following sections include a discussion of CLI commands that can be used to configure resource parameters and monitor or reset statistics for switch resources. These commands include:

- **health threshold**—Configures threshold limits for input traffic (RX), output/input traffic (TX/RX), memory usage**,** CPU usage, and chassis temperature. See page 36-41 for more information.
- **show health configuration**—Displays current health threshold settings. See page 36-42 for details.
- **health interval**—Configures sampling interval between health statistics checks. See page 36-43 for more information..
- **show health** —Displays health statistics for the switch, as percentages of total resource capacity. See page 36-44 for more information.

Configuring Resource Thresholds

Health Monitoring software monitors threshold levels for the switch's consumable resources—*bandwidth, RAM memory, and CPU capacity*. When a threshold is exceeded, the Health Monitoring feature sends a trap to the Network Management Station (NMS). A trap is an alarm alerting the user to specific network events. In the case of health-related traps, a specific indication is given to determine which threshold has been crossed.

Note. When a resource falls back below the configured threshold, an addition trap is sent to the user. This indicates that the resource is no longer operating beyond its configured threshold limit.

The **health threshold** command is used to configure threshold limits for input traffic (RX), output/input traffic (TX/RX), memory usage and CPU usage.

To configure thresholds for these resources, enter the **health threshold** command, followed by the input traffic, output/input traffic, memory usage, or CPU usage where:

rx	Specifies an **input traffic** (RX) threshold, in percentage. This value defines the maximum percentage of total bandwidth allowed for *incoming traffic only*. The total bandwidth is the Ethernet port capacity of *all NI modules* currently operating in the switch, in Mbps. For example, a chassis with 48 100Base-T Ethernet ports installed has a total bandwidth of 4800 Mbps. Since the default RX threshold is 80 percent, the threshold is exceeded if the input traffic on all ports reaches 3840 Mbps or higher.
txrx	Specifies a value for the **output/input traffic** (TX/RX) threshold. This value defines the maximum percentage of total bandwidth allowed for *all incoming and outgoing traffic*. As with the RX threshold described above, the total bandwidth is defined as the Ethernet port capacity for all NI modules currently operating in the switch, in Mbps. The default TX/RX threshold is 80 percent.
memory	Specifies a value for the **memory usage** threshold. Memory usage refers to the total amount of RAM memory currently used by switch applications. The default memory usage threshold is 80 percent.
cpu	Specifies a value for the **CPU usage** threshold. CPU usage refers to the total amount of CPU processor capacity currently used by switch applications. The default CPU usage threshold is 80 percent.

For example, to specify a CPU usage threshold of 85 percent, enter the following command:

```
-> health threshold cpu 85
```

For more information on the **health threshold** command, refer to Chapter 52, "Health Monitoring Commands," in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Note. When you specify a new value for a threshold limit, the value is automatically applied across all levels of the switch (switch, module, and port). You cannot select differing values for each level.

Displaying Health Threshold Limits

The **show health configuration** command is used to view all current health thresholds on the switch, as well as individual thresholds for input traffic (RX), output/input traffic (TX/RX), memory usage and CPU usage.

To view all health thresholds, enter the following command:

```
-> show health configuration
Rx Threshold            = 80,
TxRx Threshold          = 80,
Memory Threshold        = 80,
CPU Threshold           = 80,
Sampling Interval (Secs)= 10
```

Note. For detailed definitions of each of the threshold types, refer to "Configuring Resource Thresholds" on page 36-41, as well as Chapter 52, "Health Monitoring Commands," in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Sampling Intervals

The **sampling interval** is the period of time between polls of the switch's consumable resources to monitor performance vis-a-vis previously specified thresholds. The **health interval** command can be used to configure the sampling interval between health statistics checks.

To configure the sampling interval, enter the **health interval** command, followed by the number of seconds.

For example, to specify a **sampling interval** value of 6 seconds, enter the following command:

```
-> health interval 6
```

Valid values for the seconds parameter include 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, or 30.

Note. If the sampling interval is decreased, switch performance be affected.

Viewing Sampling Intervals

The **show health** command can be used to display the current health sampling interval (period of time between health statistics checks), measured in seconds.

To view the sampling interval, enter the **show health configuration** command. The currently configured health sampling interval (measured in seconds) is displayed, as shown below:

```
-> show health configuration

Rx Threshold            = 80,
TxRx Threshold          = 80,
Memory Threshold        = 80,
CPU Threshold           = 80,
Sampling Interval (Secs) = 10
```

Viewing Health Statistics for the Switch

The **show health** command can be used to display health statistics for the switch.

To display health statistics, enter the **show health** command, followed by the slot/port location.

For example, to view health statistics for the entire switch, enter the **show health** command without specifying any additional parameters. A screen similar to the following example is displayed, as shown below:

```
-> show health
* - current value exceeds threshold

Device                            1 Min  1 Hr  1 Hr
Resources          Limit   Curr   Avg    Avg   Max
----------------+-------+------+------+-----+----
Receive              80     00     00    00    00
Transmit/Receive     80     00     00    00    00
Memory               80     87*    87    86    87
Cpu                  80     08     05    04    08
Temperature Cmm      60     34     34    33    34
Temperature Cmm Cpu  60     28     28    27    28
```

In the screen sample shown above, the Device Resources field displays the device resources that are being measured (for example, Receive displays statistics for traffic received by the switch; Transmit/Receive displays statistics for traffic transmitted and received by the switch; Memory displays statistics for switch memory; and CPU displays statistics for the switch CPU). The Limit field displays currently configured device threshold levels as percentages of available bandwidth. The Curr field displays current bandwidth usage for the specified device resource. 1 Min. Avg. refers to the average device bandwidth used over a 1 minute period. 1 Hr. Avg. refers to the average device bandwidth used over a 1 hour period, and 1 Hr. Max. refers to the maximum device bandwidth used over a 1 hour period.

Note. If the Current value appears with an asterisk displayed next to it, the Current value exceeds the Threshold limit. For example, if the Current value for Memory is displayed as 85* and the Threshold Limit is displayed as 80, the asterisk indicates that the Current value has exceeded the Threshold Limit value.

Viewing Health Statistics for a Specific Interface

To view health statistics for slot 4/port 3, enter the **show health** command, followed by the appropriate slot and port numbers. A screen similar to the following example is displayed, as shown below:

```
-> show health port 1/4/3
* - current value exceeds threshold

Port 1/4/3                              1 Min     1 Hr      1 Hr
Resources            Limit    Curr     Avg       Avg       Max
------------------+----------+--------+--------+---------+--------------------
Receive              80       01       01       01         01
Receive/Transmit     80       01       01       01         01
```

In the screen sample shown above, the port 04/03 Resources field displays the port resources that are being measured (for example, Receive displays statistics for traffic received by the switch, while Transmit/Receive displays statistics for traffic transmitted and received by the switch). The Limit field displays currently configured resource threshold levels as percentages of available bandwidth. The Curr field displays current bandwidth usage for the specified resource. 1 Min. Avg. refers to the average resource bandwidth used over a 1 minute period. 1 Hr. Avg. refers to the average resource bandwidth used over a 1 hour period, and 1 Hr. Max. refers to the maximum resource bandwidth used over a 1 hour period.

37 Configuring VLAN Stacking

VLAN Stacking provides a mechanism to tunnel multiple customer VLANs (CVLAN) through a service provider network using one or more service provider VLANs (SVLAN) by way of 802.1Q double-tagging or VLAN Translation. This feature enables service providers to offer their customers Transparent LAN Services (TLS). This service is multipoint in nature so as to support multiple customer sites or networks distributed over the edges of a service provider network.

This implementation of VLAN Stacking offers the following functionality:

- Ingress bandwidth sharing across User Network Interface (UNI) ports.
- Ingress bandwidth rate limiting on a per UNI port, per CVLAN, or CVLAN per UNI port basis.
- CVLAN (inner) tag 802.1p-bit mapping to SVLAN (outer) tag 802.1p bit.
- CVLAN (inner) tag DSCP mapping to SVLAN (outer) tag 802.1p bit.
- Profiles for saving and applying traffic engineering parameter values.

In This Chapter

This chapter describes the basic components of VLAN Stacking and how to define a service-based or port-based configuration through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

This chapter provides an overview of VLAN Stacking and includes the following topics:

- "VLAN Stacking Defaults" on page 37-2.
- "VLAN Stacking Overview" on page 37-3.
- "Interaction With Other Features" on page 37-7.
- "Configuring VLAN Stacking Services" on page 37-10.
- "VLAN Stacking Application Example" on page 37-27.
- "Wire-Rate Hardware Loopback Test" on page 37-30.
- "Verifying the VLAN Stacking Configuration" on page 37-35.

VLAN Stacking Defaults

Parameter Description	Command	Default Value/Comments
SVLAN administrative and Spanning Tree status.	**ethernet-service svlan**	Enabled
Vendor TPID and legacy BPDU support for STP on a VLAN Stacking network port.	**ethernet-service nni**	TPID = 0x8100 legacy STP BPDU = dropped.
Acceptable frame types on a VLAN Stacking user port.	**ethernet-service sap cvlan**	None.
Traffic engineering profile attributes for a VLAN Stacking Service Access Point (SAP).	**ethernet-service sap-profile**	ingress bandwidth = shared ingress bandwidth mbps = 0 CVLAN tag is preserved. SVLAN priority mapping = 0
Treatment of customer protocol control frames ingressing on a VLAN Stacking user port.	**ethernet-service uni-profile**	Processed Frames: UDLD, OAM, LACPMarker Tunneled Frames: 802.1ab, 802.3ad, STP, MVRP, Discarded Frames: VTP, VLAN, Uplink Fast, PVST, PAGP, DTP, CDP

VLAN Stacking Overview

VLAN Stacking provides a mechanism for defining a transparent bridging configuration through a service provider network. The major components of VLAN Stacking that provide this type of functionality are described as follows:

- **Provider Edge (PE) Bridge—**An ethernet switch that resides on the edge of the service provider network. The PE Bridge interconnects customer network space with service provider network space. A switch is considered a PE bridge if it transports packets between a customer-facing port and a network port or between two customer-facing ports.
- **Transit Bridge—**An ethernet switch that resides inside the service provider network and provides a connection between multiple provider networks. It employs the same SVLAN on two or more network ports. This SVLAN does not terminate on the switch itself; traffic ingressing on a network port is switched to other network ports. It is also possible for the same switch to function as a both a PE Bridge and a Transit Bridge.
- **Tunnel (SVLAN)—**A tunnel, also referred to as an SVLAN, is a logical entity that connects customer networks by transparently bridging customer traffic through a service provider network. The tunnel is defined by an SVLAN tag that is appended to all customer traffic. This implementation provides an SVLAN that is defined by the type of traffic that it carries - an SVLAN that *carries customer traffic*.
- **Network Network Interface (NNI)—**An NNI is a port that resides on either a PE Bridge or a Transit Bridge and connects to a service provider network. Traffic ingressing on a network port is considered SVLAN traffic and is switched to a customer-facing port or to another network port.
- **User Network Interface (UNI)—**A UNI is a port that resides on a PE bridge that connects to a customer network and carries customer traffic. The UNI may consist of a single port or an aggregate of ports and can accept tagged or untagged traffic.

The following illustration shows how VLAN Stacking uses the above components to tunnel customer traffic through a service provider network:

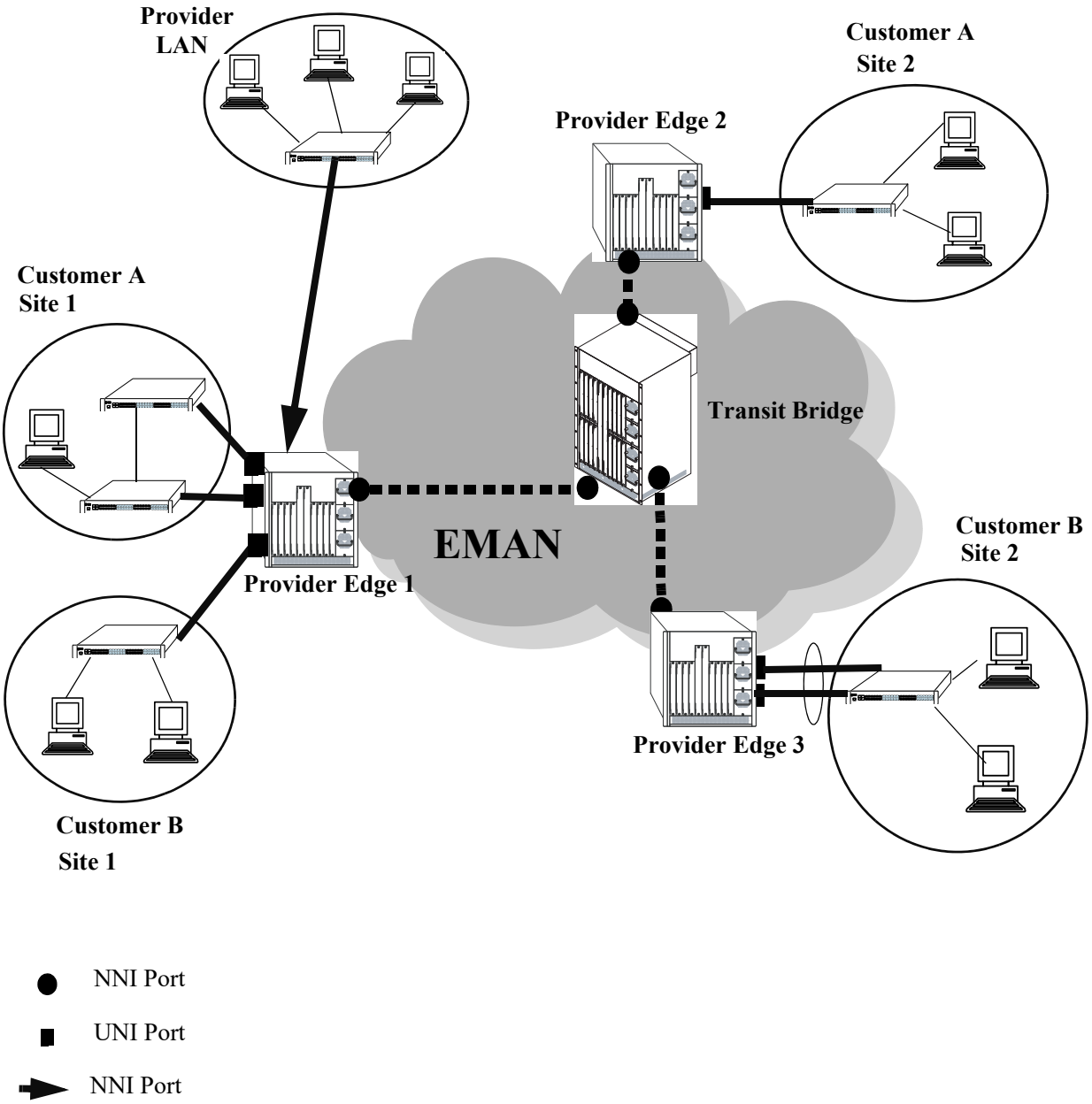


Figure 37-1 : VLAN Stacking Elements

How VLAN Stacking Works

On the Provider Edge bridge (PE), a unique tunnel (SVLAN) ID is assigned to each customer. The tunnel ID corresponds to a VLAN ID, which is created on the switch when the tunnel is configured. For example, when tunnel 100 is created, VLAN Stacking software interacts with VLAN Manager software to configure a VLAN 100 on the switch. VLAN 100 is the provider bridge VLAN that will tunnel customer VLAN traffic associated with tunnel 100. So, there is a one to one correspondence between a tunnel and its provider bridge VLAN ID. In fact, tunnel and VLAN are interchangeable terms when referring to the provider bridge configuration.

VLAN Stacking refers to the tunnel encapsulation process of appending to customer packets an 802.1Q tag that contains the tunnel ID associated to that customer's provider bridge port and/or VLANs. The encapsulated traffic is then transmitted through the Ethernet metro area network (EMAN) cloud and received on another PE bridge that contains the same tunnel ID, where the packet is then stripped of the tunnel tag and forwarded to the traffic destination.

The following provides an example of how a packet ingressing on a VLAN Stacking UNI port that is tagged with the customer VLAN (CVLAN) ID transitions through the VLAN Stacking encapsulation process:

1 Packet with CVLAN tag ingressing on a user port.

MAC DA (6)	MAC SA (6)	CVLAN Tag (4)	ETYPE 0x0800	Payload

2 **Double Tagging** inserts the SVLAN tag in the packet. The packet is sent out the network port with double tags (SVLAN+CVLAN).

MAC DA (6)	MAC SA (6)	SVLAN Tag (4)	CVLAN Tag (4)	ETYPE 0x0800	Payload

3 **VLAN Translation** replaces the CVLAN Tag with SVLAN Tag. The packet is sent out the network port with a single tag (SVLAN).

MAC DA (6)	MAC SA (6)	SVLAN Tag (4)	ETYPE 0x0800	Payload

VLAN Stacking Services

The VLAN Stacking application uses an Ethernet service based approach for tunneling customer traffic through a provider network. This approach requires the configuration of the following components to define a tunneling service:

- **VLAN Stacking Service—**A service name that is associated with an SVLAN, NNI ports, and one or more VLAN Stacking service access points. The service identifies the customer traffic that the SVLAN will carry through the provider traffic.
- **Service Access Point (SAP)—**A SAP is associated with a VLAN Stacking service name and a SAP profile. The SAP binds UNI ports and customer traffic received on those ports to the service. The profile specifies traffic engineering attribute values that are applied to the customer traffic received on the SAP UNI ports.
- **Service Access Point (SAP) Profile—**A SAP profile is associated with a SAP ID. Profile attributes define values for ingress bandwidth sharing, rate limiting, CVLAN tag processing (translate or preserve), and priority mapping (inner to outer tag or fixed value).
- **UNI Port Profile**—This type of profile is associated with each UNI port and configures how Spanning Tree and other control packets are processed on the UNI port.

See the "Configuring VLAN Stacking Services" on page 37-10 for more information.

Interaction With Other Features

This section contains important information about VLAN Stacking interaction with other OmniSwitch features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

Link Aggregation

- Both static and dynamic link aggregation are supported with VLAN Stacking.
- Note that a link aggregate must consist of all UNI or all NNI ports. VLAN Stacking functionality is not supported on link aggregates that consist of a mixture of VLAN Stacking ports and conventional switch ports.

Quality of Service (QoS)

The QoS application has the following interactions with VLAN Stacking:

- By default, QoS allocates switch resources for VLAN Stacking Service attributes, even though such attributes are not configurable via the QoS CLI.
- The **ethernet-service sap-profile** command is used to create a VLAN Stacking service access point (SAP) profile. When the **bandwidth not-assigned** and **priority not-assigned** parameters are used with this command, QoS is prevented from allocating switch resources for the SAP profile.
- VLAN Stacking ports are trusted and use 802.1p classification.
- If there is a conflict between VLAN Stacking Service attributes and the QoS configuration, the VLAN Stacking attributes are given precedence over QoS policies.
- QoS applies the **inner source vlan** and **inner 802.1p** policy conditions to the CVLAN (inner) tag of VLAN Stacking packets.
- QoS applies the **source vlan** and **802.1p** policy conditions to the SVLAN (outer) tag of VLAN Stacking packets.

Spanning Tree

- Spanning Tree is automatically enabled for VLAN Stacking SVLANs. The Spanning Tree status for an SVLAN is configurable through VLAN Stacking commands. Note that the SVLAN Spanning Tree status applies only to the service provider network topology.
- BPDU frames are tunneled by default. See "Configuring a UNI Profile" on page 37-20 for information about configuring VLAN Stacking to tunnel or discard Spanning Tree BPDU.
- See "Configuring VLAN Stacking Network Ports" on page 37-13 for information about configuring VLAN Stacking interoperability with *legacy* Spanning Tree BPDU systems.
- A back door link configuration is not supported. This occurs when there is a link between two customer sites that are both connected to a VLAN Stacking provider edge switch.
- A dual home configuration is not supported. This type of configuration consists of a single customer site connected to two different VLAN Stacking switches or two switches at a customer site connect to two different VLAN Stacking switches.

Quick Steps for Configuring VLAN Stacking

The following steps provide a quick tutorial for configuring a VLAN Stacking service:

1 Create a VLAN Stacking VLAN (SVLAN) 1001 using the **ethernet-service svlan** command.

```
-> ethernet-service svlan 1001
```

2 Create a VLAN Stacking service and associate the service with SVLAN 1001 using the **ethernet-service service-name** command.

```
-> ethernet-service service-name CustomerA svlan 1001
```

3 Configure port 1/1/3 as a VLAN Stacking Network Network Interface (NNI) port using the **ethernet-service nni** command.

```
-> ethernet-service nni port 1/1/3
```

4 Associate NNI port 1/1/3 with SVLAN 1001 using the **ethernet-service svlan nni** command.

```
-> ethernet-service svlan 1001 nni port 1/1/3
```

5 Create a VLAN Stacking Service Access Point (SAP) and associate it to the “CustomerA” service using the **ethernet-service sap** command.

```
-> ethernet-service sap 10 service-name CustomerA
```

6 Configure port 1/1/40 as a VLAN Stacking User Network Interface (UNI) port and associate the port with SAP ID 10 using the **ethernet-service sap uni** command.

```
-> ethernet-service sap 10 uni port 1/1/40
```

7 Associate traffic from customer VLANs (CVLAN) 10 and 20 with SAP 10 using the **ethernet-service sap cvlan** command.

```
-> ethernet-service sap 10 cvlan 10
-> ethernet-service sap 10 cvlan 20
```

8 (Optional) Create a SAP profile that applies an ingress bandwidth of 10, translates the CVLAN tag, and maps the CVLAN priority to the SVLAN priority using the **ethernet-service sap-profile** command.

```
-> ethernet-service sap-profile sap-video1 ingress-bandwidth 10 cvlan translate
priority map-inner-to-outer-p
```

9 (Optional) Associate the “sap-video1” profile with SAP 10 using the **ethernet-service sap sap-profile** command.

```
-> ethernet-service sap 10 sap-profile sap-video1
```

10 (Optional) Create a UNI port profile to block STP control frames received on UNI ports using the **ethernet-service uni-profile** command.

```
-> ethernet-service uni-profile uni_1 l2-protocol stp discard
```

11 (Optional) Associate the “uni_1” profile with port 1/1/40 using the **ethernet-service uni uni-profile** command.

```
-> ethernet-service uni port 1/1/40 uni-profile uni_1
```

Note. Verify the VLAN Stacking Ethernet service configuration using the **show ethernet-service** command:

```
-> show ethernet-service

Service Name : VideoOne
    SVLAN   : 300
    NNI(s)  : 1/2/1, 1/1/3
    SAP Id  : 20
        UNIs        : 1/1/35, 1/1/40
        CVLAN(s)    : 10, 20
        sap-profile : sap-video1
    SAP Id  : 30
        UNIs        : 1/1/30
        CVLAN(s)    : untagged, 40
        sap-profile : sap-video2

Service Name : CustomerABC
    SVLAN   : 255
    NNI(s)  : 1/1/22
    SAP Id  : 10
        UNIs        : 1/1/10, 1/1/11
        CVLAN(s)    : 500, 600
        sap-profile : default-sap-profile

-> show ethernet-service service-name CustomerABC

Service Name : CustomerABC
    SVLAN   : 255
    NNI(s)  : 1/1/22
    SAP Id  : 10
        UNIs        : 1/1/10, 1/1/11
        CVLAN(s)    : 500, 600
        sap-profile : default-sap-profile

-> show ethernet-service svlan 300

Service Name : VideoOne
    SVLAN   : 300
    NNI(s)  : 1/2/1, 1/1/3
    SAP Id  : 20
        UNIs        : 1/1/35, 1/1/40
        CVLAN(s)    : 10, 20
        sap-profile : sap-video1
    SAP Id  : 30
        UNIs        : 1/1/30
        CVLAN(s)    : 30, 40
        sap-profile : sap-video2
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in this display.

Configuring VLAN Stacking Services

Configuring a VLAN Stacking Ethernet service requires several steps. These steps are outlined here and further described throughout this section. For a brief tutorial on configuring a VLAN Stacking service, see "Quick Steps for Configuring VLAN Stacking" on page 37-8.

1 Create an SVLAN. An SVLAN is associated to a VLAN Stacking service to carry customer or provider traffic. See "Configuring SVLANs" on page 37-11.

2 Create a VLAN Stacking service. A service name is associated with an SVLAN to identify the customer traffic that the SVLAN will carry through the provider network. See "Configuring a VLAN Stacking Service" on page 37-12.

3 Configure Network Network Interface (NNI) ports. An NNI port is associated with an SVLAN and carries the encapsulated SVLAN traffic through the provider network. See "Configuring VLAN Stacking Network Ports" on page 37-13.

4 Configure a VLAN Stacking service access point (SAP). A SAP binds UNI ports, the type of customer traffic, and traffic engineering parameter attributes to the VLAN Stacking service. Each SAP is associated to one service name, but a single service can have multiple SAPs to which it is associated. See "Configuring a VLAN Stacking Service Access Point" on page 37-15.

5 Configure User Network Interface (UNI) ports. One or more UNI ports are associated with a SAP to identify to the service which ports will receive customer traffic that the service will process for tunneling through the provider network. When a UNI port is associated with a SAP, the SAP parameter attributes are applied to traffic received on the UNI port. See "Configuring VLAN Stacking User Ports" on page 37-16.

6 Associate CVLAN traffic with a SAP. This step specifies the type of traffic customer traffic that is allowed on UNI ports and then tunneled through the SVLAN. The type of customer traffic is associated with a SAP and applies to all UNI ports associated with the same SAP. See "Configuring the Type of Customer Traffic to Tunnel" on page 37-16.

7 Define SAP profile attributes. A SAP profile contains traffic engineering attributes for specifying bandwidth sharing, rate limiting, CVLAN translation or double-tagging, and priority bit mapping. A default profile is automatically associated with a SAP at the time the SAP is created. As a result, it is only necessary to configure a SAP profile if the default attribute values are not sufficient. See "Configuring a Service Access Point Profile" on page 37-18.

8 Define UNI profile attributes. A default UNI profile is automatically assigned to a UNI port at the time a port is configured as a VLAN Stacking UNI. This profile determines how control frames received on the port are processed. It is only necessary to configure a UNI profile if the default attribute values are not sufficient. See "Configuring a UNI Profile" on page 37-20.

The following table provides a summary of commands used in these procedures:

Commands	Used for ...
ethernet-service svlan	Creating SVLANs to tunnel customer traffic.
ethernet-service service-name	Creating a VLAN Stacking service and associating the service with an SVLAN.
ethernet-service svlan nni	Configuring a switch port as a VLAN Stacking NNI port and associating the NNI port with an SVLAN.
ethernet-service nni	Configuring a vendor TPID and legacy Spanning Tree support for an NNI port.
ethernet-service sap	Creating a VLAN Stacking SAP and associates the SAP with a VLAN Stacking service name.
ethernet-service sap uni	Configuring a switch port as a VLAN Stacking UNI port and associating the UNI port with a VLAN Stacking SAP.
ethernet-service sap cvlan	Specifying the type of customer traffic that is accepted on SAP UNI ports.
ethernet-service sap-profile	Configures traffic engineering attributes for customer traffic that is accepted on SAP UNI ports.
ethernet-service sap sap-profile	Associates a VLAN Stacking SAP with a profile.
ethernet-service uni-profile	Configures how protocol control frames are processed on VLAN Stacking UNI ports.
ethernet-service uni uni-profile	Associates a VLAN Stacking UNI port with a profile.

Configuring SVLANs

SVLANs carry customer traffic and are not configurable or modifiable using standard VLAN commands.

The **ethernet-service svlan** command is used to create an SVLAN. This command provides parameters to specify the type of SVLAN: **svlan** for customer traffic.For example, the following command creates a customer SVLAN:

```
-> ethernet-service svlan 300
```

Similar to standard VLANs, the administrative and Spanning Tree status for the SVLAN is enabled by default and the SVLAN ID is used as the default name. The **ethernet-service svlan** command also provides parameters for changing any of these status values and the name. These are the same parameters that are used to change these values for standard VLANs. For example, the following commands change the administrative and Spanning Tree status and name for SVLAN 300:

```
-> ethernet-service svlan 300 disable
-> ethernet-service svlan 300 stp disable
-> ethernet-service svlan 300 name "Customer A"
```

To delete an SVLAN from the switch configuration, use the **no** form of the **ethernet-service svlan** command. For example, to delete SVLAN 300 enter:

```
-> no ethernet-service svlan 300
```

Note that when an SVLAN is deleted, all port associations with the SVLAN are also removed.

Use the **show ethernet-service vlan** command to display a list of VLAN Stacking VLANs configured for the switch.

Configuring a VLAN Stacking Service

A VLAN Stacking service is identified by a name. The **ethernet-service service-name** command is used to create a service and assign the service to an SVLAN ID, depending on the type of traffic the service will process. The ID specified with this command identifies the SVLAN that will carry traffic for the service. Each service is associated with only one SVLAN, but an SVLAN may belong to multiple services.

To create a VLAN Stacking service, use the **ethernet-service service-name** command and specify a name and SVLAN ID. For example, the following command creates a service named "Video-Service" and associates the service with SVLAN 300:

```
-> ethernet-service service-name Video-Service svlan 300
```

The SVLAN ID specified with this command must already exist in the switch configuration; entering a standard VLAN ID is not allowed. See "Configuring SVLANs" on page 37-11 for more information.

Once the VLAN Stacking service is created, the name is used to configure and display all components associated with that service. The service name provides a single point of reference for a specific VLAN Stacking configuration. For example, the following **show ethernet-service** command display shows how the service name identifies a VLAN Stacking service and components related to that service:

```
-> show ethernet-service
Service Name : VideoOne
    SVLAN   : 300
    NNI(s)  : 1/2/1, 1/1/3
    SAP Id  : 20
        UNIs        : 1/1/35, 1/1/40
        CVLAN(s)    : 10, 20
        sap-profile : sap-video1
    SAP Id  : 30
        UNIs        : 1/1/30
        CVLAN(s)    : untagged, 40
        sap-profile : sap-video2

Service Name : CustomerABC
    SVLAN   : 255
    NNI(s)  : 1/1/22
    SAP Id  : 10
        UNIs        : 1/1/10, 1/1/11
        CVLAN(s)    : 500, 600
        sap-profile : default-sap-profile

-> show ethernet-service service-name CustomerABC
Service Name : CustomerABC
SVLAN   : 255
    NNI(s)  : 1/1/22
    SAP Id  : 10
        UNIs        : 1/1/10, 1/1/11
        CVLAN(s)    : 500, 600
        sap-profile : default-sap-profile
```

```
-> show ethernet-service svlan 300
Service Name : VideoOne
    SVLAN   : 300
    NNI(s)  : 1/2/1, 1/1/3
    SAP Id  : 20
        UNIs        : 1/1/35, 1/1/40
        CVLAN(s)    : 10, 20
        sap-profile : sap-video1
    SAP Id  : 30
        UNIs        : 1/1/30
        CVLAN(s)    : untagged, 40
        sap-profile : sap-video2
```

To delete a service from the switch configuration, use the **no** form of the **ethernet-service service-name** command. For example, the following command deletes the "Video-Service" service:

```
-> no ethernet-service service-name Video-Service
```

Note that when a VLAN Stacking service is deleted, the SVLAN ID association with the service is automatically deleted. However, if one or more VLAN Stacking service access point (SAP) are associated with the service, remove the SAPs first before attempting to delete the service.

Configuring VLAN Stacking Network Ports

The **ethernet-service nni** command is used to configure a switch port or link aggregate of ports as a VLAN Stacking Network Network Interface (NNI). For example, the following command configures port 2/1 as an NNI port:

```
-> ethernet-service nni port 1/2/1
```

When a port is converted to a NNI port, the default VLAN for the port is changed to a VLAN that is reserved for the VLAN Stacking application. At this point, the port is no longer configurable using standard VLAN port commands.

The **ethernet-service nni** command is also used to optionally specify the following parameter values that are applied to traffic proessed by the NNI port:

- **tpid—**Configures the vendor TPID value for the SVLAN tag. This value is set to the default and is applied to traffic egressing on the NNI port and is compared to the SVLAN tag of packets ingressing on the NNI port. If the configured NNI TPID value and the ingress packet value match, then the packet is considered an SVLAN tagged packet. If these values do not match, then the packet is classified as a non-SVLAN tagged packet.
- **stp legacy-bpdu—**Specifies whether or not legacy Spanning Tree BPDU are tunneled on the NNI port.

The following command example configures the vendor TPID for NNI port 2/1 to 0x88a8 and enables support for Spanning Tree legacy BPDU:

```
-> ethernet-service nni port 1/2/1 tpid 88a8 stp legacy-bpdu enable
```

Consider the following when configuring NNI port parameter values:

- A mismatch of TPID values on NNI ports that are connected together is not supported; VLAN Stacking will not work between switches using different NNI TPID values.
- Enable legacy BPDU support only on VLAN Stacking network ports that are connected to legacy BPDU switches. Enabling legacy BPDU between AOS switches may cause flooding or an unstable network.

- If legacy BPDU is enabled on a network port while at same time BPDU flooding is enabled on user ports, make sure that tagged customer BPDUs are not interpreted by intermediate switches in the provider network.
- If the peer switch connected to the VLAN Stacking network port supports the Provider MAC address (i.e., STP, 802.1ad/D6.0 MAC), then enabling legacy BPDU support is not required on the network port. Refer to the following table to determine the type of STP MAC used:

STP	
Customer MAC	{0x01, 0x80, 0xc2, 0x00, 0x00, 0x00}
Provider MAC address (802.1ad/D6.0)	{0x01, 0x80, 0xc2, 0x00, 0x00, 0x08}
Provider MAC address (Legacy MAC)	{0x01, 0x80, 0xc2, 0x00, 0x00, 0x00}
Provider MAC address	{0x01, 0x80, 0xc2, 0x00, 0x00, 0x0D}

- STP legacy BPDU are supported only when the flat Spanning Tree mode is active on the switch.
- NNI ports can be 802.1q tagged or untagged with normal VLANs. This allows the NNI port to carry both 802.1q tagged traffic and SVLAN tagged traffic, and untagged traffic. 802.1q VLAN tagging to an NNI interface will not be allowed if the interface is set with TPID other than 0x8100.

Configuring a NNI Association with an SVLAN

The **ethernet-service svlan nni** command is used to associate the NNI with an SVLAN. For example, the following command associates NNI 1/2/1 with SVLAN 300:

```
-> ethernet-service svlan 300 nni port 1/2/1
```

When a port is associated with an SVLAN using this command, the port is automatically defined as an NNI to carry traffic for the specified SVLAN.

To delete an NNI port association with an SVLAN, use the **no** form of the **ethernet-service svlan nni** command. For example, the following command deletes the NNI 2/1 and SVLAN 300 association:

```
-> no ethernet-service svlan 300 nni port 1/2/1
```

Use the **show ethernet-service nni** command to display the NNI port configuration for the switch.

Configuring a VLAN Stacking Service Access Point

The **ethernet-service sap** command is used to configure a VLAN Stacking service access point (SAP). A SAP is assigned an ID number at the time it is configured. This ID number is then associated with the following VLAN Stacking components:

- **User Network Interface (UNI) ports**. See "Configuring VLAN Stacking User Ports" on page 37-16.
- **Customer VLANs (CVLANs).** See "Configuring the Type of Customer Traffic to Tunnel" on page 37-16.
- **SAP profile.** Each SAP is associated with a single profile. This profile contains attributes that are used to define traffic engineering parameters applied to traffic ingressing on UNI ports that are associated with the SAP. See "Configuring a Service Access Point Profile" on page 37-18.

The above components are all configured separately using different VLAN Stacking commands. The **ethernet-service sap** command is for creating a SAP ID and associating the ID with a VLAN Stacking service. For example, the following command creates SAP 20 and associates it with Video-Service:

```
-> ethernet-service sap 20 service-name Video-Service
```

To delete a VLAN Stacking SAP from the switch configuration, use the **no** form of the **ethernet-service sap** command. For example, the following command deletes SAP 20:

```
-> no ethernet-service sap 20
```

Note that when the SAP is deleted, all UNI port, CVLAN, and profile associations are automatically dropped. It is not necessary to remove these items before deleting the SAP.

A VLAN Stacking SAP basically identifies the location where customer traffic enters the provider network edge, the type of customer traffic to service, parameters to apply to the traffic, and the service that will process the traffic for tunneling through the provider network.

Consider the following when configuring a VLAN Stacking SAP:

- A SAP is assigned to only one service, but a service can have multiple SAPs. So, a single service can process and tunnel traffic for multiple UNI ports and customers.
- Associating multiple UNI ports to one SAP is allowed.
- A default SAP profile is associated with the SAP at the time the SAP is created. This profile contains the following default attribute values:

Ingress bandwidth sharing	**shared**
Ingress bandwidth maximum	0
Egress bandwidth maximum	0
CLAN tag	preserve (double-tag)
Priority mapping	**fixed** 0

  The above default attribute values are applied to customer traffic associated with the SAP. Only one profile is assigned to each SAP; however, it is possible to use the same profile for multiple SAPs.

- To use different profile attribute values, create a new profile and associate it with the SAP. See "Configuring a Service Access Point Profile" on page 37-18. Each time a profile is assigned to a SAP, the existing profile is overwritten with the new one.

Use the **show ethernet-service sap** command to display the SAPs configured for the switch. Use the **show ethernet-service** command to display a list of VLAN Stacking services and the SAPs associated with each service.

Configuring VLAN Stacking User Ports

The **ethernet-service sap uni** command is used to configure a switch port or a link aggregate as a VLAN Stacking User Network Interface (UNI) and associate the UNI with a VLAN Stacking service access point (SAP). For example, the following command configures port 1/1/1 as an UNI port and associates 1/1/1 with SAP 20:

```
-> ethernet-service sap 20 uni port 1/1/1
```

A UNI port is a customer-facing port on which traffic enters the VLAN Stacking service. When the port is associated with a service access point, the port is automatically defined as a UNI port and the default VLAN for the port is changed to a VLAN that is reserved for the VLAN Stacking application.

To delete a UNI port association with a VLAN Stacking SAP, use the **no** form of the **ethernet-service sap uni** command. For example, the following command deletes the association between UNI port 1/1/1 and SAP 20:

```
-> no ethernet-service sap 20 uni port 1/1/1
```

Note that when the last SAP association for the port is deleted, the port automatically reverts back to a conventional switch port and is no longer VLAN Stacking capable.

Consider the following when configuring VLAN Stacking UNI ports:

- All customer traffic received on the UNI port is dropped until customer VLANs (CVLAN) are associated with the port. See "Configuring the Type of Customer Traffic to Tunnel" on page 37-16.
- A default UNI profile is assigned to the port at the time the port is configured. This profile defines how control frames received on the UNI ports are processed.
- To use different profile attribute values, create a new profile and associate it with the UNI port. See "Configuring a UNI Profile" on page 37-20. Each time a profile is assigned to a UNI, the existing profile is overwritten with the new one.
- Only fixed ports can be converted to UNI ports.

Use the **show ethernet-service uni** command to display a list of UNI ports and the profile association for each port.

Configuring the Type of Customer Traffic to Tunnel

The **ethernet-service sap cvlan** command is used to associate customer traffic with a VLAN Stacking service access point (SAP). This identifies the type of customer traffic received on the SAP UNI ports that the service will process and tunnel through the SVLAN configured for the service. For example, the following command specifies that traffic tagged with customer VLAN (CVLAN) 500 is allowed on UNI ports associated with SAP 20:

```
-> ethernet-service sap 20 cvlan 500
```

In this example, customer frames tagged with VLAN ID 500 that are received on SAP 20 UNI ports are processed by the service to which SAP 20 is associated. This includes applying profile attributes

associated with SAP 20 to the qualifying customer frames. If no other customer traffic is specified for SAP 20, all other frames received on SAP 20 UNI ports are dropped.

In addition to specifying one or more CVLANs, it is also possible to specify the following parameters when using the **ethernet-service sap cvlan** command:

- **all—**Specifies that all untagged and tagged frames are accepted on the UNI ports. This mapping denotes that all customer frames that do not map to any other SAP, will be mapped into this service.
- **untagged**—Specifies that only untagged frames are accepted on the UNI ports. This mapping denotes that only untagged frames will be mapped into this service.

For example, the following command specifies that all untagged frames are accepted on UNI ports associated with SAP 20:

```
-> ethernet-service sap 20 cvlan untagged
```

Use the **no** form of the **ethernet-service sap cvlan** command to delete an association between customer traffic and a VLAN Stacking SAP. For example, the following command deletes the association between CVLAN 500 and SAP 20:

```
-> no ethernet-service sap 20 cvlan 500
```

Note that when the last customer traffic association is deleted from a SAP, the SAP itself is not automatically deleted. No traffic is accepted or processed by a SAP in this state, but the SAP ID is still known to the switch.

Consider the following when configuring the type of customer traffic to tunnel:

- If no customer traffic is associated with a VLAN Stacking SAP, then the SAP does not process any traffic for the service.
- Only one **all** or **untagged** designation is allowed for any given SAP; specifying both for the same SAP is not allowed.
- Only one **untagged** designation is allowed per UNI port, even if the UNI port is associated with multiple SAPs.
- Only one **all** designation is allowed per UNI port, even if the UNI port is associated with multiple SAPs.

Use the **show ethernet-service** command to display the type of customer traffic associated with each SAP configured for the switch

Configuring a Service Access Point Profile

The **ethernet-service sap-profile** command is used to create a VLAN Stacking service access point (SAP) profile. The following command parameters define the traffic engineering attributes that are applied to customer traffic that is accepted on UNI ports associated with the SAP profile:

Profile Attribute	Command Parameters	Description
Ingress bandwidth sharing	**shared | not shared**	Whether or not ingress bandwidth is shared across UNI ports and CVLANs.
Ingress rate limiting	**ingress-bandwidth**	The rate at which customer frames ingress on UNI ports.
Egress rate limiting	**egress-bandwidth**	The rate at which customer frames egress on UNI ports.
Bandwidth assignment	**bandwidth not-assigned**	Allows QoS policy rules to override profile attribute values for bandwidth. By default, the profile bandwidth values take precedence and are allocated additional QoS system resources.
Double-tag or translate	**cvlan | preserve | translate**	Determines if a customer frame is tagged with the SVLAN ID (double-tag) or the CVLAN ID is changed to the SVLAN ID (translate) when the frame is encapsulated for tunneling. Double-tag is used by default.
Priority mapping	**map-inner-to-outer-p | map-dscp-to-outer-p | fixed**	Determines if the CVLAN (inner tag) 802.1p or DSCP value is mapped to the SVLAN (outer tag) 802.1p value or if a fixed priority value is used for the SVLAN 802.1p value. Priority mapping is set to a fixed rate of zero by default.
Priority assignment	**priority not-assigned**	Allows QoS policy rules to override profile attribute values for priority. By default, profile priority values take precedence and are allocated additional QoS system resources.

A default profile, named “default-sap-profile”, is automatically assigned to the SAP at the time the SAP is created (see “Configuring a VLAN Stacking Service Access Point” on page 37-15). It is only necessary to create a new profile to specify different attribute values if the default profile values (see above) are not sufficient.

The following command provides an example of creating a new SAP profile to specify a different method for mapping the SVLAN priority value:

```
-> ethernet-service sap-profile map_pbit priority map-inner-to-outer-p
```

In this example the **map_pbit** profile specifies priority mapping of the CVLAN inner tag 802.1p value to the SVLAN outer tag value. The other attributes in this profile are set to their default values.

To delete a SAP profile, use the **no** form of the **ethernet-service sap-profile** command. For example, the following command deletes the **map_pbit** profile:

```
-> no ethernet-service sap-profile map_pbit
```

Consider the following when configuring a SAP profile:

- When a profile is created, **bandwidth not-assigned** and **priority not-assigned** parameters are *not* specified. This means that even if no bandwidth value is specified or the priority is set to fixed, QoS still allocates switch resources to enforce bandwidth and priority settings for the profile. In addition, QoS policy rules cannot override the profile bandwidth or priority settings.
- Use the **bandwidth not-assigned** and **priority not-assigned** parameters to prevent the profile from triggering QoS allocation of switch resources. When a profile is created using these parameters, QoS policy rules/ACLs are then available to define more custom bandwidth and priority settings for profile traffic. For example, mapping several inner DSCP/ToS values to the same outer 802.1p value.
- Egress bandwidth can be configured only for SVLANs.
- A CVLAN-UNI combination associated with a SAP having egress bandwidth configuration is unique and it cannot be configured on any other SAP with egress bandwidth configuration.

Use the **show ethernet-service sap-profile** command to view a list of profiles that are already configured for the switch. This command also displays the attribute values for each profile.

Associating a Profile with a Service Access Point

After a profile is created, it is then necessary to associate the profile with a VLAN Stacking SAP. When this is done, the current profile associated with a SAP is replaced with the new profile.

The **ethernet-service sap sap-profile** command is used to associate a new profile with a VLAN Stacking SAP. For example, the following command associates the map_pbit profile to SAP 20:

```
-> ethernet-service sap 20 sap-profile map_pbit
```

To change the profile associated with the SAP back to the default profile, specify “default-sap-profile” for the profile name. For example:

```
-> ethernet-service sap 20 sap-profile default-sap-profile
```

Use the **show ethernet-service sap** command to display the SAP configuration, which includes the profile association for each SAP.

Configuring a UNI Profile

A UNI port profile determines how control frames ingressing on a VLAN Stacking UNI port are processed. When a port is configured as a UNI port, a default Layer 2 profile (**def-uni-profile**) is applied to the port with default values for how to process Layer 2 control frames.

If the default profile values are not sufficient, use the **ethernet-service uni-profile** command to create a new UNI port profile. For example, the following command creates a UNI profile named "uni_1" to specify that VLAN Stacking should discard MVRP frames:

```
-> ethernet-service uni-profile uni_1 l2-protocol mvrp discard
```

Consider the following when configuring Layer 2 profiles:

- Not all of the protocol parameters are currently supported with the **peer**, **tunnel**, and **discard** parameters. Refer to the **ethernet-service uni-profile** command page in the *OmniSwitch AOS Release 8 CLI Reference Guide* for a table that shows the supported profile actions for each protocol parameter.
- When a profile is created, the new profile inherits the default profile settings for processing control frames. The default settings are then applied with the new profile unless they are explicitly changed. For example, the profile "uni_1" was configured to discard MVRP frames. No other protocol settings were changed, so the default settings still apply for the other protocols.
- A UNI profile cannot be modified or deleted if it is associated with a UNI port. Delete all associations with any UNI ports before attempting to modify or delete a UNI profile.

To delete a UNI profile, use the **no** form of the **ethernet-service uni-profile** command. For example, the following command deletes the "uni_1" profile:

```
-> no ethernet-service uni-profile uni_1
```

Use the **show ethernet-service uni-profile** command to display a list of profiles (including the default profile) that are configured for the switch.

Configuring UNI Profile Action for 802.1AB PDUs

A UNI profile can be configured to apply a different action to tagged and untagged 802.1AB control frames. For example, the following command uses the **ethernet-service uni-profile inbound 802.1ab** command with the **tagged** parameter to set the action for tagged 802.1AB control frames in the specified UNI profile:

```
-> ethernet-service uni-profile lldp-tagged inbound tagged l2-protocol 802.1ab
tunnel
```

In this example, the tunnel action is specified only for tagged 802.1AB control frames. All tagged 802.1AB control frames will be tunneled, while untagged frames will be dropped.

To configure an action for untagged 802.1AB control frames, use the **ethernet-service uni-profile inbound 802.1ab** command with the **untagged** parameter to set the action for untagged 802.1AB control frames in the specified UNI profile. For example:

```
-> ethernet-service uni-profile lldp-untagged inbound untagged l2-protocol
802.1ab peer
```

In this example, the peer action is specified only for untagged 802.1AB control frames. All untagged 802.1AB control frames will participate in the protocol, while tagged frames will be dropped.

To configure the same action for both tagged and untagged 802.1AB control frames, use the **ethernet-service uni-profile inbound 802.1ab** command with the **both** parameter. For example:

```
-> ethernet-service uni-profile lldp-both inbound both l2-protocol 802.1ab peer
```

When a UNI profile is configured to apply a different action for tagged and untagged 802.1AB PDUs, the profile action can only be modified through one of the following methods:

- Set both the tagged and untagged action for 802.1AB PDUs back to the default setting (**discard**) then configure a new action for both.
- Delete the UNI profile and create a new one with the modified action for tagged and untagged 802.1AB PDUs.

Configuring a Destination MAC Address

The **ethernet-service uni-profile** command can also be used to configure a destination MAC address that is used to tunnel L2 protocol packets through the provider network. When the MAC tunneling action is specified for a protocol, the destination MAC address for the protocol is changed to the destination MAC address associated with the UNI profile.

By default, the destination MAC address for the profile is set to 01:00:0c:cd:cd:d0. To change this to a different address, use the **ethernet-service uni-profile** command with the **tunnel-mac** option. For example:

```
-> ethernet-service uni-profile uni_1 tunnel-mac 01:00:0c:cd:cd:cd
```

To specify a MAC tunneling action for a protocol, use the **ethernet-service uni-profile** command with the **mac-tunnel** option. For example, the following command configures the VTP protocol to use the profile destination MAC address instead of the protocol destination MAC address:

```
-> ethernet-service uni-profile uni_1 l2-protocol vtp mac-tunnel
```

When VTP control packets are processed by the “uni-1” profile, the destination MAC address for the packet is changed to the destination MAC address associated with the “uni-1” profile.

Associating UNI Profiles with UNI Ports

After a UNI profile is created, it is then necessary to associate the profile with a UNI port or a UNI link aggregate. When this is done, the current profile associated with the port is replaced with the new profile.

The **ethernet-service uni uni-profile** command is used to associate a new profile with a UNI port. For example, the following command associates the “uni_1” profile to UNI port 1/1/1:

```
-> ethernet-service uni port 1/1/1 uni-profile uni_1
```

To change the profile associated with the UNI port back to the default profile, specify “default-uni-profile” for the profile name. For example:

```
-> ethernet-service uni port 1/1/1 uni-profile default-uni-profile
```

Associating IEEE UNI Profiles with UNI Ports

In addition to the default UNI profile, there are also two built-in UNI profiles (*ieee-fwd-all* and *ieee-drop-all*) that can be assigned to a UNI port. Use one of the profiles to tunnel or discard all IEEE multicast MAC address traffic received on the UNI port.

– When a UNI port is assigned to the *ieee-fwd-all* profile, all L2 protocol control packets destined for 01:80:C2:00:00:XX are forwarded as normal data. However, control packets with a destination

MAC address of 01:80:C2:00:00:01, 01:80:C2:00:00:04, or 01:80:C2:00:00:08 are not forwarded end-to-end.

– When a UNI port is assigned to the *ieee-drop-all* profile, all L2 protocol control packets destined for 01:80:C2:00:00:XX are discarded.

– When a UNI port is assigned to either one of the built-in profiles (*ieee-fwd-all* or *ieee-drop-all*), tunneled L2 protocol control packets (tagged packets with SVLAN ID) received on NNI ports are forwarded as normal data.

To assign one of the built-in IEEE UNI profiles to a UNI port, use the **ethernet-service uni uni-profile** command and specify the built-in profile name. For example:

```
-> ethernet-service uni port 1/1/6 uni-profile ieee-fwd-all
-> ethernet-service uni port 1/1/7 uni-profile ieee-drop-all
```

Use the **show ethernet-service uni** command to display the profile associations for each UNI port.

Configuring a Custom L2 Protocol

A custom L2 protocol entry is configured to define a proprietary protocol with a destination multicast MAC address and is assigned to a UNI profile for specific packet control. The **ethernet-service custom-L2-protocol** command is used to configure a custom L2 protocol entry. For example, the following command creates a custom L2 protocol with the name “p1” and MAC address 01:80:c2:00:11:11 associated to the custom-L2-protocol:

```
-> ethernet-service custom-L2-protocol p1 mac 01:80:c2:00:11:11
```

A custom L2 protocol can be configured with the following values:

When configuring a custom L2 protocol with ...	The ...
a MAC address and no mask	MAC address cannot be: • a reserved IPv4/IPv6 multicast address. • a MAC-specific control protocol (01-80-C2-00-00-01 or 01-80-C2-00-00-04). • a Service OAM address (01-80-C2-00-00-30 to 01-80-C2-00-00-3F). • used in another custom L2 protocol without a mask.
a MAC address with a mask	MAC address range cannot overlap with: • reserved IPv4/IPv6 multicast address ranges. • a MAC address range configured for another custom L2 protocol. Only nested ranges are allowed.
an EtherType and optional Sub-Type	The EtherType/Sub-Type value cannot be: • configured if a mask was specified for the MAC address. • configured for another custom L2 protocol. • a well-known L2 protocol ((0x8809/1, 0x8809/2, 0x8809/3, 0x888E, 0x88CC, 0x88F5).
an optional SSAP/DSAP and PID	The SSAP/DSAP PID value cannot be: • configured if a mask or EtherType value was specified for the MAC address. • configured for another custom L2 protocol.

Use the **show ethernet-service custom-l2-protocol** command to view the configuration information for a custom L2 protocol entry.

Associating a Custom L2 Protocol with a UNI Profile

A custom L2 protocol is associated to a UNI profile for specific packet control actions (tunnel, MAC tunnel, and discard).

- The tunnel and discard actions apply to all traffic matching a custom L2 protocol entry.
- The MAC tunnel action applies only to traffic matching a custom L2 protocol entry that was configured with an EtherType and optional Sub-Type value or an SSAP/DSAP PID value.

The following table describes the supported packet control actions:

Action	Description
Tunnel	Tunnels packets with a destination MAC address that matches the MAC address configured for the custom L2 protocol across the provider network without modifying the MAC address.
MAC tunnel	Changes the destination MAC address of a packet to the configured tunnel MAC address for the specified UNP profile.
Discard	Discards packets with a destination MAC address that matches the MAC address configured for the custom L2 protocol.

To associate a a custom L2 protocol with a UNI profile, use the **ethernet-service uni-profile custom-L2-protocol** command with the **tunnel**, **mac-tunnel**, or **discard** option. For example, the following command specifies the action “mac-tunnel” for the custom L2 protocol “tunnel-mac-ethertype” associated with the UNI profile “profile1”:

```
-> ethernet-service uni-profile profile1 custom-L2-protocol tunnel-mac-ethertype
mac-tunnel
```

Use the **show ethernet-service uni-profile** command to display the custom L2 protocol entries assigned to a UNI profile.

Control Protocol Tunneling Frame Statistics

The following tunneling protocol statistics are collected:

- **RX frame statistics at UNI port level**: On a per-UNI port and per-protocol basis, the following information is collected:
  - The number of frames received and trapped for processing.
  - The number of frames that are peered and MAC tunneled.
  - The source MAC address of the last frame received on each port for each protocol.
- **TX frame statistics at UNI port level**: On a per-UNI port and per-protocol basis, this statistic indicates the number of frames that are MAC de-tunneled and transmitted on the UNI port.
- **RX frame statistics at NNI port level**: On a per-NNI port basis, this statistic indicates the number of frames that were trapped for processing because their destination MAC address matched the configured tunnel MAC address and the number of trapped frames discarded.
- **RX frame statistics at UNI profile level**: This statistic indicates the number of protocol frames (including custom L2 protocols) that are received on all UNI ports that are bound to a UNI profile.

The following subsections describe the statistic types and the CLI commands that are used to display the statistics.

UNI Port Statistics

UNI port statistics are statistics of frames trapped for processing. The frames are processed if the action applied to the frame is MAC tunnel or peer. In addition, frames received on an NNI port are processed if the frames need to be de-tunneled.

The **show ethernet-service uni l2pt-statistics** command is used to display the following UNI port statistics:

- The number of UNI frames received on UNI ports trapped for processing.
- The number of UNI frames trapped for each action (peer and MAC tunnel).
- The number of de-tunneled frames before they are flooded on the UNI port.
- The source MAC address of the last frame received on each port for each protocol.

To clear UNI port statistics, use the **clear ethernet-service uni l2pt-statistics** command. Statistics are also cleared when one of the following occurs:

- The UNI port association with a UNI profile is removed. When this happens, the port is automatically assigned to the default UNI profile and the statistics collected are based on the default UNI profile.
- The UNI port association with a UNI profile is changed. Statistics are then collected going forward based on the new UNI profile.
- The port is no longer configured as a UNI port.

NNI Port Statistics

NNI port statistics provide the number of frames trapped on an NNI port and the number of frames discarded. The **show ethernet-service nni l2pt-statistics** command is used to display the following NNI port statistics:

- The number of frames received on NNI ports and trapped for processing.
- The number of frames discarded because they do not match any de-tunnel criteria for trapping NNI frames for processing.

To clear NNI port statistics, use the **clear ethernet-service nni l2pt-statistics** command. Statistics are also cleared when the port is no longer configured as an NNI port.

UNI Profile Statistics

Statistics are collected for protocol frames (including custom L2 protocols) that are received on all ports that are bound to the UNI profile. The protocol frames included in these statistics are those trapped for processing with a drop or MAC tunnel action.

The **show ethernet-service uni-profile l2pt-statistics** command is used to display the following UNI profile statistics:

- The total number of frames received for the UNI profile.
- The total number of frames received for each protocol for the UNI profile.
- The actual treatment applied for each L2 protocol.

To clear UNI profile statistics, use the **clear ethernet-service uni-profile l2pt-statistics** command. Statistics are also cleared when one of the following occurs:

- The UNI profile is modified or deleted.
- There is a change to the port configuration for the UNI profile (for example, a UNI port association is added or removed).

Configuring MAC Tunneling for an SVLAN

The status of MAC tunneling can be configured on a global or per-SVLAN basis.

- When MAC tunneling is enabled globally, all Generic Bridge PDU Tunneling (GBPT) packets are trapped for processing even if there is no MAC tunneling action configured for the associated UNI profile. This limits the rate at which these packets are forwarded.
- Enabling MAC tunneling for specific SVLANs limits the trapping of GBPT packets to only those SVLANs where it is needed for MAC tunneling; other SVLANs forward the traffic without trapping the GBPT packets.

The following sections describe the various configurations required to activate this functionality.

Global MAC Tunneling Status

When globally enabled (the default), MAC tunneling is active for all UNI profile protocols that are configured with the MAC tunneling action. To enable or disable the global MAC tunneling status, use the **ethernet-service mac-tunneling** command. For example:

```
-> ethernet-service mac-tunneling enable
-> ethernet-service mac-tunneling disable
```

Notes:

- When MAC tunneling is enabled globally, per-SVLAN MAC tunneling configuration will not be active.
- When MAC tunneling is disabled globally, the MAC tunnel status of the SVLANs configured will be active.

SVLAN MAC Tunneling Status

Enabling MAC tunneling for specific SVLANs limits the trapping of GBPT packets for processing to only those SVLANs. To configure the MAC tunneling status on a per SVLAN basis, use the **ethernet-service svlan mac-tunneling** command. For example:

```
-> ethernet-service svlan 1000 mac-tunneling enable name "VLAN 1000"
-> ethernet-service svlan 1000 mac-tunneling disable name "VLAN 1000"
```

Note. Enabling MAC tunneling on a per-SVLAN basis or on a global basis is mutually exclusive; make sure MAC tunneling is globally disabled before attempting to enable MAC tunneling for the specified SVLAN. To view the status of MAC tunneling for an SVLAN, use the **show vlan** command with the VLAN ID parameter.

Transparent Bridging

Transparent bridging associates NNI ports with all possible VLANs even if they are not created on the switch. The OmniSwitch can support this by having the administrator create all possible VLANs and associate them to NNI ports. However, transparent bridging has an advantage over the conventional configuration approach by reducing the administrative effort of configuring all VLANs and their associated VPAs.

Note. Transparent bridging is supported only when the switch is running in the flat (RSTP) Spanning Tree mode; it is not supported in the per-VLAN mode.

Transparent bridging associates all VLANs to the specified NNI port and Spanning Tree group one. This feature is typically limited to a ring topology where there are only two NNI ports/link aggregates per switch. Transparent bridging must be enabled both globally and per port or linkagg using the **ethernet-service transparent-bridging** command, for example:

```
-> ethernet-service transparent-bridging enable
-> ethernet-service nni port 1/1/5 transparent-bridging enable
-> ethernet-service nni linkagg 5 transparent-bridging enable
```

Use the **show ethernet-service** command to display transparent bridging configuration.

VLAN Stacking Application Example

The VLAN Stacking feature provides the ability to transparently connect multiple customer sites over a single shared service provider network. This section demonstrates this ability by providing a sample VLAN Stacking configuration that tunnels customer VLANs (CVLAN) inside a service provider VLAN (SVLAN} so that customer traffic is transparently bridged through a Metropolitan Area Network (MAN).

The illustration below shows the sample VLAN Stacking configuration described in this section. In this configuration, the provider edge bridges will encapsulate Customer A traffic (all CVLANs) into SVLAN 100 and Customer B traffic (CVLAN 10 only) into SVLAN 200. In addition, the CVLAN 10 inner tag priority bit value is mapped to the SVLAN out tag priority value. The customer traffic is then transparently bridged across the MAN network and sent out to the destined customer site.

Double-tagging is the encapsulation method used in this application example, This method consists of appending the SVLAN tag to customer packets ingressing on provider edge UNI ports so that the traffic is bridged though the provider network SVLAN. The SVLAN tag is then stripped off of customer packets egressing on provider edge UNI ports before the packets are delivered to their destination customer site.

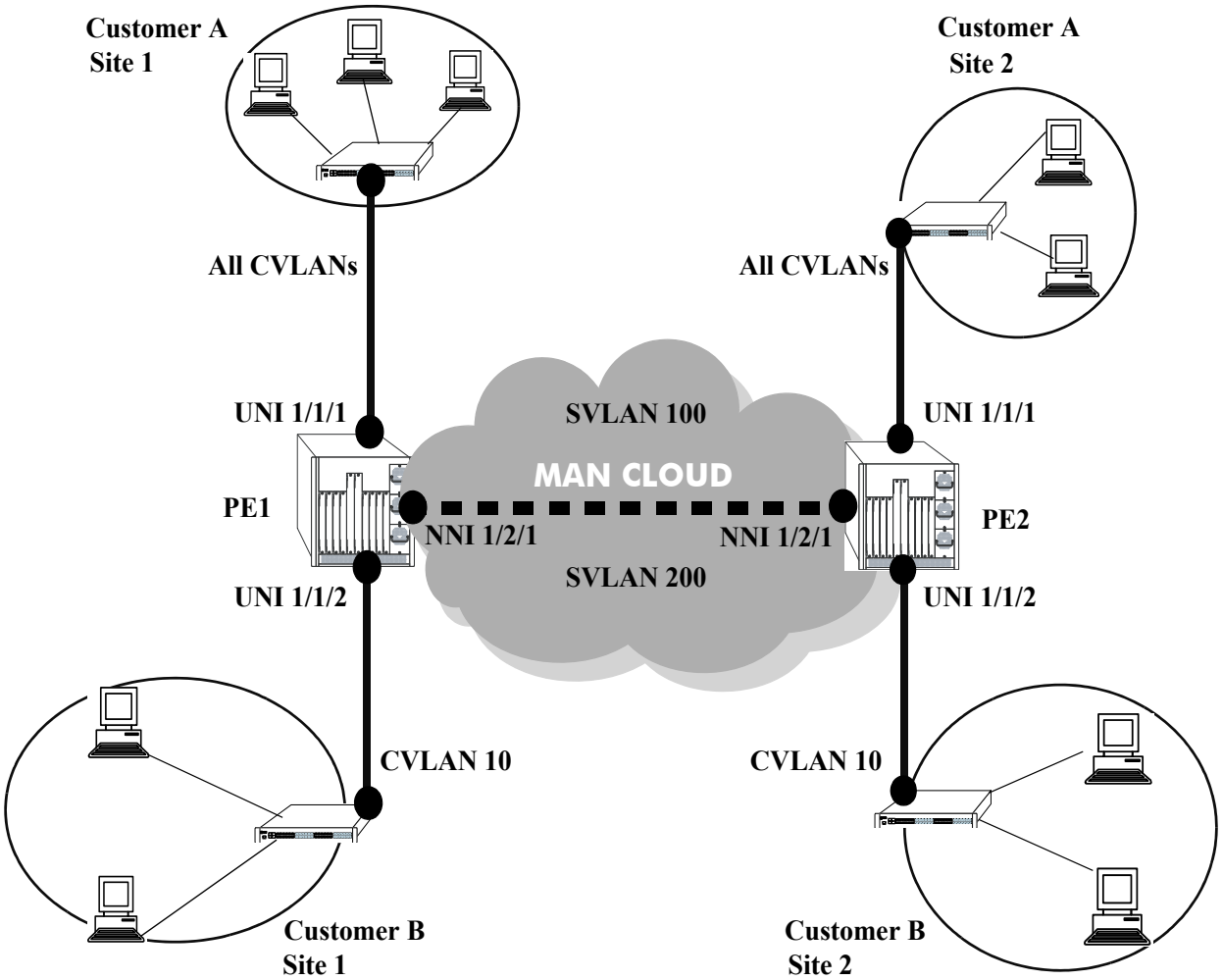


Figure 37-2 : VLAN Stacking Application

VLAN Stacking Configuration Example

This section provides a tutorial for configuring the sample application, as illustrated on page 37-27, using VLAN Stacking Ethernet services. This tutorial assumes that both provider edge switches (PE1 and PE2) are operating in the VLAN Stacking service mode.

1 Configure SVLAN 100 and SVLAN 200 on PE1 *and* PE2 switches using the **ethernet-service svlan** command.

```
-> ethernet-service svlan 100
-> ethernet-service svlan 200
```

2 Configure two VLAN Stacking services on PE1 *and* PE2 using the **ethernet-service service-name** command. Configure one service with the name “CustomerA” and the other service with the name “Customer B”. Assign “CustomerA” service to SVLAN 100 and “CustomerB” service to SVLAN 200.

```
-> ethernet-service service-name CustomerA svlan 100
-> ethernet-service service-name CustomerB svlan 200
```

3 Configure port 1/2/1 on PE1 *and* PE2 as VLAN Stacking NNI ports using the **ethernet-service svlan nni** command. Associate each port with both SVLAN 100 and SVLAN 200.

```
-> ethernet-service svlan 100 nni port 1/2/1
-> ethernet-service svlan 200 nni port 1/2/1
```

4 Configure a VLAN Stacking SAP with ID 20 on PE1 *and* PE2 using the **ethernet-service sap**. Associate the SAP with the “CustomerA” service.

```
-> ethernet-service sap 20 service-name CustomerA
```

5 Configure a VLAN Stacking SAP with ID 30 on PE1 *and* PE2 using the **ethernet-service sap** command. Associate the SAP with the “CustomerB” service.

```
-> ethernet-service sap 30 service-name CustomerB
```

6 Configure port 1/1/1 on PE1 *and* PE2 as a VLAN Stacking UNI port and associate 1/1/1 with SAP 20 using the **ethernet-service sap uni** command.

```
-> ethernet-service sap 20 uni port 1/1/1
```

7 Configure port 1/1/2 on PE1 *and* PE2 as a VLAN Stacking UNI port and associate 1/1/2 with SAP 30 using the **ethernet-service sap uni** command.

```
-> ethernet-service sap 30 uni port 1/1/2
```

8 Configure SAP 20 on PE1 *and* PE2 to accept all customer traffic using the **ethernet-service sap cvlan** command.

```
-> ethernet-service sap 20 cvlan all
```

9 Configure SAP 30 on PE1 *and* PE2 to accept only customer traffic that is tagged with CVLAN 10 using the **ethernet-service sap cvlan** command.

```
-> ethernet-service sap 30 cvlan 10
```

10 Create a SAP profile on PE1 *and* PE2 that will map the inner CVLAN tag 802.1p value to the outer SVLAN tag using the **ethernet-service sap-profile** command.

```
-> ethernet-service sap-profile map_pbit priority map-inner-to-outer-p
```

11 Associate the "map_pbit" profile to SAP 30 using the **ethernet-service sap sap-profile** command. This profile will only apply to Customer B traffic, so it is not necessary to associate the profile with SAP 20.

```
-> ethernet-service sap 30 sap-profile map_pbit
```

12 Verify the VLAN Stacking service configuration using the **show ethernet-service** command.

```
-> show ethernet-service

Service Name : CustomerA
    SVLAN   : 100
    NNI(s)  : 1/2/1
    SAP Id  : 20
        UNIs        : 1/1/1
        CVLAN(s)    : all
        sap-profile : default-sap-profile

Service Name : CustomerB
    SVLAN   : 200
    NNI(s)  : 1/2/1
    SAP Id  : 30
        UNIs        : 1/1/2
        CVLAN(s)    : 10
        sap-profile : map_pbit

-> show ethernet-service service-name CustomerB

Service Name : CustomerB
    SVLAN   : 200
    NNI(s)  : 1/2/1
    SAP Id  : 30
        UNIs        : 1/1/2
        CVLAN(s)    : 10
        sap-profile : map_pbit
```

The following is an example of what the sample configuration commands look like entered sequentially on the command line of the provider edge switches:

```
-> ethernet-service svlan 100
-> ethernet-service service-name CustomerA svlan 100
-> ethernet-service svlan 100 nni port 1/2/1
-> ethernet-service sap 20 service-name CustomerA
-> ethernet-service sap 20 uni 1/1/1
-> ethernet-service sap 20 cvlan all

-> ethernet-service svlan 200
-> ethernet-service service-name CustomerB svlan 200
-> ethernet-service svlan 200 nni port 1/2/1
-> ethernet-service sap 30 service-name CustomerB
-> ethernet-service sap 30 uni 1/1/2
-> ethernet-service sap 30 cvlan 10
-> ethernet-service sap-profile map_pbit priority map-inner-to-outer-p
-> ethernet-service sap 30 sap-profile map_pbit
```

Wire-Rate Hardware Loopback Test

A wire-rate hardware loopback test function is available to perform In-Service and Out-of-Service throughput testing during initial turn-up or on-the-fly in an active network. The loopback tests can be used to validate the configured Service Level Agreements (SLAs) and QoS parameters that are associated with a service or a flow.

The loopback test capability provided allows the use of an external test head to send traffic at wire-rate speed to a specific switch port which then loops the traffic back to the test head. The test head measures and collects statistics on frame loss, delay, and latency of the loopback traffic.

Two types of loopback tests are supported with this implementation:

- Inward loopback
- Outward loopback

The inward test loops back test frames ingressing on a given port. The outward test loops back test frames egressing on a given port.

Configuring an Ethernet Loopback Test

The type of loopback test performed is determined by a user-configured test profile that specifies the following information:

- The name of the test profile.
- The destination MAC address for the test frame must be unique to the network and must not be used anywhere in the device.
- The VLAN ID on which the test frames are forwarded. Always use the outer VLAN ID.
- The switch port (for example, the UNI or NNI port) that performs the egress or ingress loopback operation for the test.
- The type of test to run (outward or inward loopback).

The **loopback-test** command is used to define the test profile and is also the same command that is used to enable or disable the actual loopback operation. For example, the following command creates an inward loopback test profile:

```
-> loopback-test PE1-inward-UNI destination-mac 00:00:00:cc:aa:bb port 1/1/2
source-mac 00:00:00:dd:aa:01 vlan 1001
```

The following command creates an egress loopback test profile:

```
-> loopback-test PE2-outward-UNI destination-mac 00:00:00:cc:ab:bb port 1/1/2
source-mac 00:00:00:dd:ab:01 vlan 1001 type outward
```

The following commands enable and disable the **PE1-inward-UNI** profile attributes for the switch:

```
-> loopback-test PE1-inward-UNI admin-state enable
-> loopback-test PE1-inward-UNI admin-state disable
```

Use the **show loopback-test** command to display the loopback test profile configuration.

Configuration Guidelines (OmniSwitch 6860, 6865)

Consider the following guidelines when configuring an Ethernet loopback test on an OmniSwitch 6860 or OmniSwitch 6865:

- Up to eight profiles are configurable per switch.
- Only Layer 2 loopback tests are supported, so test frames are not routed. As a result, the loopback test operation will only swap the source and destination MAC address of bridged test frames.
- Test frame must be L3 frames and L3 header must be included in test frames.
- The destination MAC address configured in the outward loopback test profile will be learned as the static MAC address on the loopback port. All traffic with this destination MAC address and VLAN will be forwarded to the loopback port.
- On outward loopback test port, all non-test frame traffic is dropped.
- Each loopback test is associated with one VLAN; using multiple VLANs is not supported.
- Ports used for an outward loopback operation go “out-of-service” and no longer carry customer traffic. The port does remain active, however, for test frame traffic.
- Ports used for an inward loopback operation remain “in-service”. Test frame traffic is mixed in with customer frame traffic.
- The port specified for an inward loopback test is the port on which test frames are received and looped back. The port specified for an outward test is the egress destination port on which the test frames are looped back. The loopback operation performed on the specified port swaps the source and destination MAC address of the test frame and then forwards the frame back to the test head.

Configuration Guidelines (OmniSwitch 6465)

Consider the following guidelines when configuring an Ethernet loopback test on an OmniSwitch 6465:

- The **loopback-test admin-state enable** command is saved to the switch configuration file. This allows a test to automatically start after the switch reboots.
- A maximum of eight loopback profiles can be configured, but only 2 profiles can be enabled at any given time.
- Defining an inward loopback test profile requires specifying only the destination MAC address and loopback port.
- Defining an outward loopback test profile requires specifying the destination MAC address, VLAN ID, and loopback port.
- A SAP ID can be specified for outward loopback test profiles; not supported with inward loopback tests profile. See “Configuring an Outward Loopback Test with a SAP ID” on page 37-32.
- If the loopback port is administratively down, then an outward loopback test cannot be started.
- The test frame does not have to be an L3 frame; both L2 and L3 test frames are supported.
- The hardware loopback feature and CPE Test Head feature can both be operational at the same time However, Test-OAM and hardware loopback cannot be configured on the same port.

Configuring an Outward Loopback Test with a SAP ID

An option to specify a SAP ID is available when configuring an outward loopback test. For example:

```
-> loopback-test PE2-outward-UNI destination-mac 00:00:00:cc:ab:bb port 1/1/2
source-mac 00:00:00:dd:ab:01 vlan 1001 type outward sap 6
```

Specifying a SAP ID uniquely identifies the SAP to use for the test. This is especially important since multiple SAPs can be associated with the same UNI port. If a SAP ID is not specified for an outward UNI or NNI loopback test, then the SAP with the lowest ID will be used.

Consider the following guidelines for specifying a SAP ID:

- The SAP ID is used to identify the bandwidth and not the CVLAN.
- Traffic for all the CVLANs that are associated with the SVLAN will get looped back, as the loopback test cannot identify which CVLAN traffic to loop back.
- Any unwanted traffic forwarded on the SVLAN will also get looped back.
- When configuring an outward NNI loopback test for a specific SAP ID, make sure the CVLAN traffic sent maps only to the matching configured SAP ID on the switch. For example, if SAP ID 10 is configured in translate mode and SAP ID 20 is configured in preserve mode and the loopback test specifies SAP ID 20, makes sure the CVLAN traffic sent maps to SAP ID 20.

Outward (Egress) Loopback Test

An outward loopback test loops back test frames egressing on a specific port. The source and destination MAC addresses of the test frames are swapped and the frames are redirected back to the port on which they were initially received and learned (the redirect port).

This type of test renders the loopback port “out-of-service”, which means the port is no longer available to forward customer traffic. Although customer frames are dropped, the port does remain in an up state and is active for looping back test frames.

An outward loopback operation is configured and performed on a UNI or NNI port. This allows to measure the performance of the customer Ethernet service as if the test frames were sent from the customer premise. The test frames would be subjected to the QoS treatment at both UNI port and NNI port.

The following illustration shows an example of an outward loopback test operation in which the loopback operation is configured on a UNI port of a provider edge switch.

Note. Conducting an outward loopback test disrupts the flow of customer traffic on the port and can cause network reachability problems.

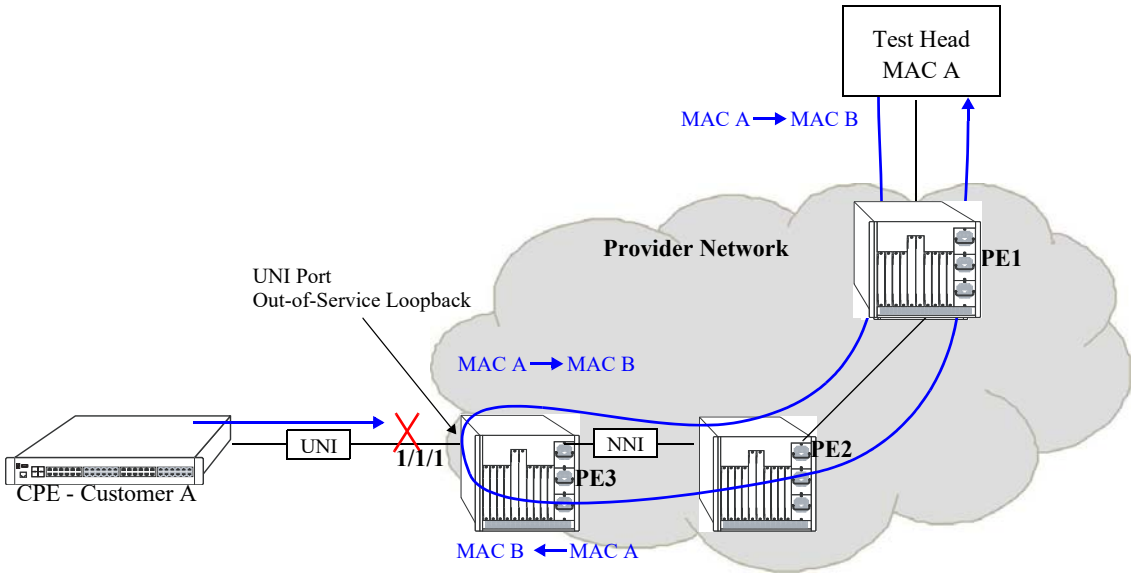


Figure 37-3 : Outward (Egress) Loopback Test Example

In this outward loopback test example:

- An outward loopback test profile is configured and enabled for UNI port 1/1/1 on PE3. The source MAC address for the profile is that of the test head (MAC A); the destination MAC address is the unique MAC address learned as the static MAC address on the configured UNI port (MAC B).
- UNI port 1/1/1 on PE3 is out of service for customer traffic.
- The test head transmits frames with source MAC A and destination MAC B.
- When the test frames reach UNI port 1/1/1 on PE3, the egress loopback operation is triggered on that port. MAC A and B are swapped in each test frame as the frames are looped back on to the egress port.

- Once the egress loopback operation is complete, the frames are sent to the redirect port and forwarded back to the test head.

Inward (Ingress) Loopback Test

An inward loopback test loops back test frames ingressing on a specific port. The source and destination MAC addresses of the test frames are swapped and the frames are redirected back to the same port. In other words, the ingress port is both the loopback and redirect port.

This type of test allows the ingress loopback port to remain “in-service” for customer traffic. As a result, customer frames and test frames are both serviced on the loopback port; there is no disruption to customer traffic.

The following illustration shows an example of an inward loopback test operation in which the loopback operation is configured on an NNI port of a provider edge switch.

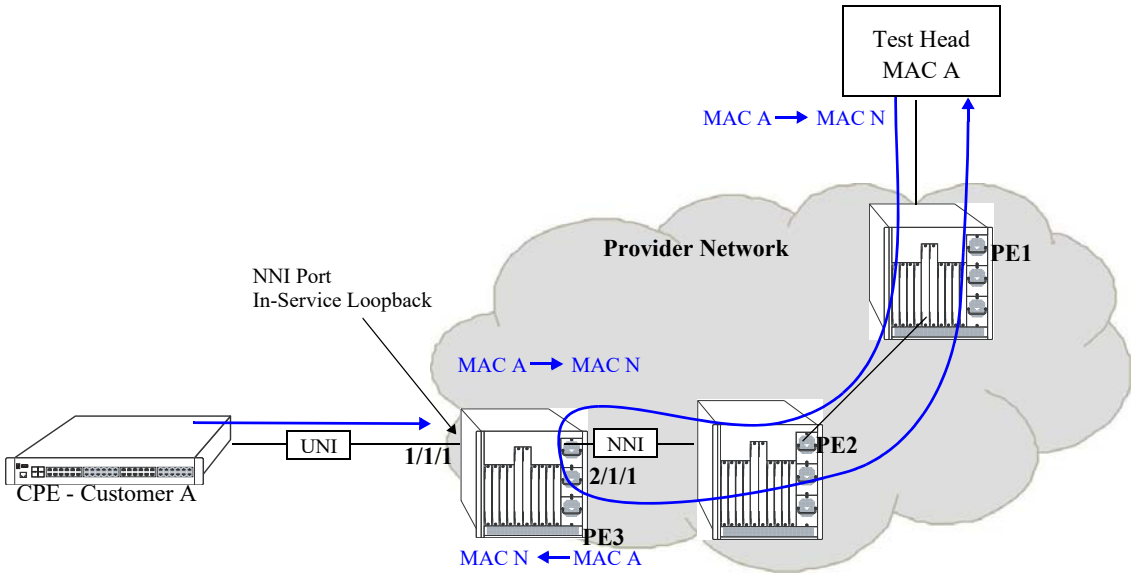


Figure 37-4 : Inward (Ingress) Loopback Test

In this inward loopback example:

- An inward loopback test profile is configured and enabled for NNI port 2/1/1 on PE3. The source MAC address for the profile is that of the test head (MAC A); the destination MAC address is the unique MAC address for PE3 (MAC N).
- NNI port 2/1/1 on PE3 is in-service for customer traffic and test frames.
- The test head transmits frames with source MAC A and destination MAC N.
- When the test frames reach NNI port 2/1/1 on PE3, the ingress loopback operation is triggered on that port. MAC A and MAC N are swapped in each test frame as the frames are looped back onto the ingress port.
- Once the ingress loopback operation is complete and since the NNI port is also the redirect port in this case, the frames are forwarded back to the test head.

Verifying the VLAN Stacking Configuration

You can use CLI **show** commands to display the current configuration and statistics of service-based VLAN Stacking on a switch. These commands include the following:

ethernet-service transparent-bridging	Displays the SVLAN configuration for the switch.
show ethernet-service	Displays the VLAN Stacking service configuration for the switch.
show ethernet-service sap	Displays the VLAN Stacking service access point (SAP) configuration for the switch.
show ethernet-service nni	Displays configuration information for NNI port parameters.
show ethernet-service uni	Displays profile associations for UNI ports.
show ethernet-service uni-profile	Displays UNI profile attribute values.
show ethernet-service sap-profile	Displays SAP profile attribute values.
show loopback-test	Displays the profile configuration for a loopback test profile.

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. An example of the output for the **show ethernet-service** command is also given in "Quick Steps for Configuring VLAN Stacking" on page 37-8.

38 Using Switch Logging

Switch logging is an event logging utility that is useful in maintaining and servicing the switch. Switch logging uses a formatted string mechanism to either record or discard event data from switch applications. The log records are copied to the output devices configured for the switch. Log records can be sent to a text file and written into the flash file system. The log records can also be scrolled to the console of the switch or to a remote IP address.

Switch logging information can be customized and configured through Command Line Interface (CLI) commands, WebView, and SNMP. Log information can be helpful in resolving configuration or authentication issues, as well as general switch errors.

This chapter describes the switch logging feature, how to configure it and display switch logging information through the Command Line Interface (CLI). CLI commands are used in the configuration examples. For more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

In This Chapter

The following procedures are described:

- "Enabling Switch Logging" on page 38-4
- "Setting the Switch Logging Severity Level" on page 38-4
- "Specifying the Switch Logging Output Device" on page 38-5
- "Displaying Switch Logging Records" on page 38-9
- "Specifying the Switch Logging Format" on page 38-11
- "Switch Logging Notifications" on page 38-11
- "Specifying the Switch Logging Record Storage Limit" on page 38-11

Note. Switch logging commands are not intended for use with low-level hardware and software debugging. It is strongly recommended that you contact an Alcatel-Lucent Enterprise Customer Service representative for assistance with debugging functions.

Switch Logging Defaults

The following table shows switch logging default values.

Global Switch Logging Defaults

Parameter Description	CLI Command	Default Value/Comments
Enabling/Disabling switch logging	**swlog**	Enabled
Switch logging severity level	**swlog appid**	Default severity level is info. The numeric equivalent for info is 6
Enabling/Disabling switch logging Output	**swlog output**	Flash Memory and Console
Enabling/Disabling uploading of the console log(s) to the remote syslog server.	**swlog output console**	Disable
Switch logging file size	**swlog output flash-file-size**	1250K bytes
Event logging file size	**show log events output**	1250K bytes

Quick Steps for Configuring Switch Logging

1 Enable switch logging by using the following command:

```
-> swlog
```

2 Specify the ID of the application to be logged along with the logging severity level.

```
-> swlog appid bridge level warning
```

Here, the application ID specifies bridging and the severity is set to the “warning” level.

3 Specify the output device to which the switch logging information must be sent.

```
-> swlog output console
-> show swlog
-> show swlog
Operational Status                  : Running,
File Size per file                  : 1250 Kbytes,
Log Device 1                        : console flash,
Syslog FacilityID                   : local0(16),
Hash Table entries age limit        : 60 seconds,
Switch Log Preamble                 : Enabled,
Switch Log Debug                    : Disabled,
Switch Log Duplicate Detection      : Enabled,
Console Display Level               : info,
RFC5424 Format Logging              : Disabled,
Swlog Threshold                     : 90 percent,
Swlog over TLS                      : Disabled
```

Switch Logging Overview

Switch logging uses a formatted string mechanism to process log requests from switch applications. When a log request is received, switch logging compares the severity level included with the request to the severity level stored for the application ID. If there is a match, a log message is generated using the format specified by the log request and placed in the switch log queue. Switch logging then returns control back to the calling application.

You can specify the path to where the log file is printed in the flash file system of the switch. You can also send the log file to other output devices, such as the console or remote IP address. In this case, the log records generated are copied to all configured output devices.

Switch logging information can be displayed and configured through CLI commands, WebView, and SNMP. The information generated by switch logging can be helpful in resolving configuration or authentication issues, as well as general errors.

Note. Although switch logging provides complementary functionality to switch debugging facilities, the switch logging commands are not intended for use with low-level hardware and software debugging functions.

Switch Logging Commands Overview

This section describes the switch logging CLI commands, for enabling or disabling switch logging, displaying the current status of the switch logging feature, and displaying stored log information.

Enabling Switch Logging

The **swlog** command initializes and enables switch logging.

To enable switch logging, enter the **swlog enable** command:

```
-> swlog enable
```

To disable switch logging, enter the **swlog disable** command:

```
-> swlog disable
```

No confirmation message appears on the screen for either command.

Setting the Switch Logging Severity Level

The switch logging feature can log all switch error-type events for a particular switch application. You can also assign severity levels to the switch applications that cause some of the events to be filtered out of your display. The **swlog appid** command is used to assign the severity levels to the applications.

The syntax for the **swlog appid** command requires that you identify a switch application and assign it a severity level. The severity level controls the kinds of error-type events that are recorded by the switch logging function. If an application experiences an event equal to or greater than the severity level assigned to the application, the event is recorded and forwarded to the configured output devices. You can specify the application either by the application ID CLI keyword or by its numeric equivalent.

To obtain a list of application IDs and numeric equivalents, use the **show swlog** command with the **appid all** parameters to display all available registered applications.

The **level** keyword is used with the **swlog appid** command to assign the error-type severity level to the specified application IDs. Values range from 1 (highest severity) to 8 (lowest severity). The values are defined in the following table:

Severity Level	Type	Description
0	**Off**	Disabled
1 *(highest severity)*	**Alarm**	A serious, non-recoverable error has occurred and the system must be rebooted.
2	**Error**	System functionality is reduced.
3	**Alert**	A violation has occurred.
4	**Warning**	An unexpected, non-critical event has occurred.
5	**event**	A clear readable customer event.
6 *(default)*	**Info**	Any other non-debug message.
7	**Debug 1**	A normal event debug message.
8	**Debug 2**	A debug-specific message.

Severity Level	Type	Description
9 *(lowest severity)*	**Debug 3**	A maximum verbosity debug message.

Specifying the Severity Level

To specify the switch logging severity level, use the **swlog appid** command. The application ID can be expressed by using the application ID keyword. The severity level can be expressed by using either the severity level number or the severity level type as shown in the table above along with the subapp. The following syntax assigns the "warning" severity level (or 5) to the "mvrpNi" subapp 1, (ID number 114) by using the severity level and application names.

```
-> swlog appid mvrpni subapp main level warning
```

The following command makes the same assignment by using the severity level and application numbers.

```
-> swlog appid mvrpni subapp 1 level 5
```

No confirmation message appears on the screen for either command.

Disabling the Severity Level

To disable the switch logging severity level, **swlog appid** command, including the application ID and severity level values. The following is a typical example:

```
-> swlog appid mvrpni subapp 1 disable
```

Or, alternatively, as:

```
-> swlog appid mvrpni subapp main disable
```

No confirmation message appears on the screen.

Specifying the Switch Logging Output Device

The **swlog output** command allows you to send the switch logging information to your console, to the switch's flash memory, or to a specified IP or IPv6 address(es).

Enabling/Disabling Switch Logging Output to the Console

To enable the switch logging output to the console, enter the following command:

```
-> swlog output console
```

To disable the switch logging output to the console, enter the following command:

```
-> no swlog output console
```

No confirmation message appears on the console screen for either command.

Enabling/Disabling Switch Logging Output to Flash Memory

To enable the switch logging output to flash memory, enter the following:

```
-> swlog output flash
```

To disable the switch logging output to flash memory, enter the following command:

```
-> no swlog output flash
```

No confirmation message appears on the screen for either command.

Specifying an IP Address for Switch Logging Output

To specify a particular IP address destination (e.g., a server) for switch logging output, enter the **swlog output** command, specifying the target IP address to which output is sent. For example, if the target IP address is 168.23.9.100, you would enter:

```
-> swlog output socket ipaddr 168.23.9.100
```

No confirmation message appears on the screen.

Disabling an IP Address from Receiving Switch Logging Output

To disable all configured output IP addresses from receiving switch logging output, enter the following command:

```
-> no swlog output socket
```

No confirmation message appears on the screen.

To disable a specific configured output IP address from receiving switch logging output, use the same command as above but specify an IPv4 or IPv6 address. For example:

```
-> no swlog output socket 174.16.5.1
```

Enabling/Disabling Uploading of Console Log to Remote Syslog Server

To enable uploading of the console log(s) to the remote UDP syslog server, enter the following command. By default, uploading the console log to a remote syslog server is disabled.

```
-> swlog output socket console enable
```

To enable uploading of console logs to a remote syslog server, it is mandatory to have at least one remote syslog server is configured (**swlog output socket** <*ip_address*>). If there is no UDP remote syslog server configured, an error message will be displayed.

To disable uploading of the console log(s) to the remote syslog server, enter the following command.

```
-> swlog output socket console disable
```

Enabling/Disabling Host Logging to Remote Syslog Server

To enable the host to send its log to the remote syslog server, enter the following command:

```
-> swlog host output socket enable
```

To enable host logging to a remote syslog server, it is mandatory to have at least one remote syslog server configured (**swlog output socket** <*ip_address*>).

To disable the host from sending the logs to the remote syslog server, enter the following command:

```
-> swlog host output socket disable
```

When all the configured remote syslog server is removed, the status of all the host remote logging is automatically set to disabled.

Enabling/Disabling NI Logging to Remote Syslog Server

To enable the NI to send its log to the remote syslog server, enter the following command:

```
-> swlog ni slot 1/3 output socket enable
```

The chassis and the slot ID of the NI from which the logs will be sent must be specified.

To enable NI logging to a remote syslog server, it is mandatory to have at least one remote syslog server configured (**swlog output socket** <*ip_address*>).

To disable the NI from sending the logs to the remote syslog server, enter the following command:

```
-> swlog ni slot 2/3 output socket disable
```

When all the configured remote syslog server is removed, the status of all the NI remote logging is automatically set to disabled.

Note. The Host Logging and NI Logging applies to OS9900 only.

Configuring syslog over TLS

Allows to send swlog to external syslog server over TLS.

To configure syslog over TLS, use the **swlog output** command. For example:

```
-> swlog output socket 192.168.120.140 tls
```

This enables syslog over TLS for the configured IP address.

Note. When syslog over TLS is enabled:

- Remote command log will not work.
- VRF cannot be used to access the syslog server.
- On the OmniSwitch 9900 only, CMM swlog is transferred to the external syslog server over TLS.
- Dying Gasp syslog messages are not captured in syslog over TLS.

To disable the syslog over TLS, enter the following command:

```
-> no swlog output socket 192.168.120.140
```

To view the configuration status of syslog over TLS, use the **show swlog** command.

Configuring the Switch Logging File Size

To configure the size of the switch logging file, use the **swlog output flash-file-size** command. To use this command, enter **swlog output flash file size** followed by the number of bytes. (The maximum size the file can be is dependent on the amount of free memory available in flash.)

For example, to set the switch logging file to 500000 bytes enter:

```
-> swlog output flash file-size 500000
```

Clearing the Switch Logging Files

You can clear the data stored in the switch logging files by executing the following command:

```
-> swlog clear
```

This command causes the switch to clear all the switch logging information and begin recording again. As a result, the switch displays a shorter file when you execute the **show log swlog** command. You want to use **swlog advanced** when the switch logging display is too long due to some of the data being old or out of date. However, the **swlog clear** command only clears the contents of the switch log files. The event logs are not cleared.

To clear both the contents and event log of the switch log file use the following command:

```
-> swlog clear all
```

No confirmation message appears on the screen.

Displaying Switch Logging Records

The **show log swlog** command can produce a display showing *all* the switch logging information or you can display information according to session, timestamp, application ID, or severity level. For details, refer to the *OmniSwitch AOS Release 8 CLI Reference Guide*. The following sample screen output shows a display of all the switch logging information.

Note. Switch logging frequently records a very large volume of data. It can take several minutes for all the switch logging information to scroll to the console screen.

```
-> show log swlog
Displaying file contents for 'swlog2.log'
FILEID: fileName[swlog2.log], endPtr[32]
                       configSize[64000], currentSize[64000], mode[2]
Displaying file contents for 'swlog1.log'
FILEID: fileName[swlog1.log], endPtr[395]
                       configSize[64000], currentSize[64000], mode[1]

Time Stamp               Application    Level   Log Message
------------------------+--------------+-------+-------------------------------
MON NOV 11 12:42:11 2005         SYSTEM    info Switch Logging files cleared by
command
MON NOV 11 13:07:26 2005            WEB    info The HTTP session login successful!
MON NOV 11 13:18:24 2005            WEB    info The HTTP session login successful!
MON NOV 11 13:24:03 2005         TELNET    info New telnet connection, Address,
128.251.30.88
MON NOV 11 13:24:03 2005         TELNET    info Session 4, Created
MON NOV 11 13:59:04 2005            WEB    info The HTTP session user logout
successful!
```

When the switch is in ASA enhanced mode, both user name and password is prompted to view the SWLOG data using **show log swlog** command. Only those users who provide the valid ASA credentials are allowed to view the data. For more information on Authenticated Switch Access - Enhanced Mode mode, refer chapter Managing Switch Security in *OmniSwitch AOS Release 8 Switch Management Guide*.

For example,

```
-> show log swlog
Username: test
Password:   *****
```

show log swlog | grep error and **show log swlog | grep more** commands are not supported in enhanced mode.

Readable Customer Event Logs

OmniSwitch is now designed to provide Readable Customer Event information about important events on the Switch in a user-friendly, consistent and customer readable format.

Use the following CLI commands to view Readable Customer Events.

Use the **swlog appid** command with level *event* to filter switch logging information for events.

```
-> swlog appid all supapp all level event
```

To display customer event logs, enter the following command.

```
-> show log events
2019 Apr 28 19:17: 8.83  : CMM : ChassisSupervisor : chassisTrapsAlert - CERTIFY w/
FLASH SYNCHRO process started
2019 Apr 28 19:17:32.697 : CMM : ChassisSupervisor : chassisTrapsAlert - CERTIFY
process completed successfully
2019 Apr 28 19:21:33.154 : CMM : ChassisSupervisor : chassisTrapsAlert - ACTIVATE
process scheduled
```

The log output is in the following format:

<SWLOG TIMESTAMP> : <CMM>/<NI> : <MODULE_NAME> : <LOG_DESCRIPTION>

To capture all event log to a file name, use the **show log events output** command. For example:

```
-> show log events output /flash/myevents
```

All the logs relating to customer events will be appended “CUSTLOG” to the prefix to differentiate events from normal debug logs.

For more information on important events and their description, see the “System Events” section in Appendix B, “SNMP Trap Information” of the *OmniSwitch AOS Release 8 Switch Management Guide*.

Specifying the Switch Logging Format

The **swlog advanced** command allows you to enable switch logging in RFC5424 format. By default, the switch logs the messages in BSD syslog format (RFC3164) to files and remote syslog servers.

When switch logging RFC5424 format is enabled, the old RFC3164 syslog messages are reformatted to comply with the RFC5424 before writing to files or sending to remote syslog servers.

The sample RFC 5424 format of swlog messages is as follows:

```
2016-10-11T13:21:00+09:30 6900 swlog - - - AAA Switch-Access debug1(6) In aaaSnap-
shot:3540, base: 20480, index: 20506
```

To enable switch logging in RFC5424 format, enter the following command:

```
-> swlog advanced enable
```

To disable switch logging in RFC5424 format, enter the following command.

```
-> swlog advanced disable
```

To view the status of the RFC5424 format, use the **show swlog** command.

Switch Logging Notifications

OmniSwitch generates swlog failure traps with a message indicating the reason of failure.

The following types of failure trigger the swlog notification:

- Fail to store swlog message to /flash/swlog files
- Fail to send swlog message to external syslog server
- Fail to configure external syslog server
- Swlog files get overwritten (old swlog messages will be lost)

Specifying the Switch Logging Record Storage Limit

The **swlog size-trap-threshold** command allows you to set the threshold limit for the storage space used to store swlog records. When the swlog record storage reaches the threshold limit, it is displayed in the swlog message. This allows to monitor the storage space and ensure the storage space is available.

For example, to set the swlog record storage limit to 70 (valid range 50 to 90), enter the following command:

```
-> swlog size-trap-threshold 70
```

To monitor the swlog threshold, use the **show swlog** command. The threshold is displayed when the set storage limit is reached.

To clear the swlog, use the **swlog clear** command.

39 Configuring Ethernet OAM

The rise in the number of Ethernet service instances has resulted in service providers requiring a powerful and robust set of management tools to maintain Ethernet service networks. Service provider networks are large and intricate, often consisting of different operators that work together to provide the customers with end-to-end services. The challenge for the service providers is to provide a highly available, convergent network to the customer base. Ethernet OAM (Operations, Administration, and Maintenance) provides the detection, resiliency, and monitoring capability for end-to-end service guarantee in an Ethernet network.

In This Chapter

This chapter describes the Ethernet OAM feature, how to configure it and display Ethernet OAM information through the Command Line Interface (CLI). For more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following information and procedures are included in this chapter:

- "Ethernet OAM Overview" on page 39-3.
- "Elements of Service OAM" on page 39-3.
- "Fault Management" on page 39-5.
- "Performance Monitoring" on page 39-5.
- "Interoperability with ITU-T Y.1731" on page 39-7.
- "Configuring Ethernet OAM" on page 39-9.
- "CVLANs for EVC MEPs" on page 39-15
- "Verifying the Ethernet OAM Configuration" on page 39-17.

For information about configuring Ethernet OAM Service Assurance Agent (SAA), see Chapter 43, "Configuring Service Assurance Agent.".

Ethernet OAM Defaults

The following table shows Ethernet OAM default values.

Parameter Description	Command	Default Value/Comments
MHF value assigned to a MD	**ethoam domain mhf**	none
ID-permission value for MD entry	**ethoam domain id-permission**	none
MHF value assigned to a MA	**ethoam association primary vlan**	defer
Continuity Check Message interval for the MA	**ethoam association ccm-interval**	10 seconds
Default domain level	**ethoam default-domain level**	0
Default domain MHF value	**ethoam default-domain mhf**	none
Default domain ID permission	**ethoam default-domain id-permission**	none
The administrative status of the MEP	**ethoam endpoint admin-state**	disable
The priority value for CCMs and LTMs transmitted by the MEP	**ethoam endpoint priority**	7
The lowest priority fault alarm for the lowest priority defect for a MEP	**ethoam endpoint lowest-priority-defect**	mac-rem-err-xcon
The priority value to be included in the inner tag of the CFM frames originating from CVLAN MEPs.	**ethoam endpoint ctag-priority**	copy-outer-to-inner
Number of Loopback messages	**ethoam loopback**	1
Fault notification generation reset time	**ethoam fault-reset-time**	1000 centiseconds

Ethernet OAM Overview

Ethernet OAM focuses on two main areas that service providers require the most and are rapidly evolving in the standards bodies:

- Service OAM (IEEE 802.1ag and ITU-T Y.1731)—for monitoring and troubleshooting end-to-end Ethernet service instances.
- Link OAM (IEEE 802.3ah EFM Link OAM)—for monitoring and troubleshooting individual Ethernet links.

These two protocols are both unique and complimentary. For example, Service OAM may isolate a fault down to a specific service, but to determine exactly where the fault occurred within the network infrastructure might also require the use of Link OAM.

Ethernet Service OAM

Ethernet Service OAM allows service providers to manage customer services end-to-end on a per-service-instance basis. A customer service instance, or Ethernet Virtual Connection (EVC), is the service that is sold to a customer and is designated by a VLAN tag on the User-to-Network Interface (UNI).

Elements of Service OAM

- Maintenance End Points (MEPs) and Maintenance Intermediate Points (MIPs)
  - MEPs initiate OAM commands. MEPs prevent leakage between domains.
  - MIPs passively receive and respond to OAM frames.
- Virtual MEP: creates an UP MEP on a virtual port.
- Maintenance Association (MA) is a logical connection between two or more MEPs.
- Point-to-point MA: logical sub-MA component only between two MEPs MA.
- Maintenance Domain: One or more MAs under the same administrative control.
- Maintenance Domain Levels: There are eight levels defined in 802.1ag:
  - levels [5, 6, 7] are for customers
  - levels [3, 4] are for service provider
  - levels [0, 1, 2] are for operators

  Multiple levels are supported for flexibility.
- Mechanisms: continuity check (CC), loopback, link trace
- Remote Fault Propagation (RFP): Propagates connectivity fault events into the interface attached to a MEP.

CFM Maintenance Domain

CFM uses a hierarchical Maintenance Domain (MD) infrastructure to manage and administer Ethernet networks.

- Each domain is made up of Maintenance Endpoints (MEPs) and Maintenance Intermediate Points (MIPs).
- The MEPs are configured on edge ports within the domain for each EVC. The MIPs are configured on relevant ports within the domain itself (interior ports).
- The network administrator selects the relevant points within the network to determine where maintenance points are needed. The maintenance point configuration defines the MD.
- MDs are assigned an unique level number (between 0 and 7) to help identify and differentiate the MD within the domain hierarchy. For example, different organizations, such as operators (levels 0, 1, 2), service providers (levels 3, 4), and customers (levels 5, 6, 7), are involved in a Metro Ethernet Service.
- Each organization can have its own Maintenance Domain, designated by the assigned level number to specify the scope of management needed for that domain.

The following illustration shows an example of the CFM Maintenance Domain hierarchy:

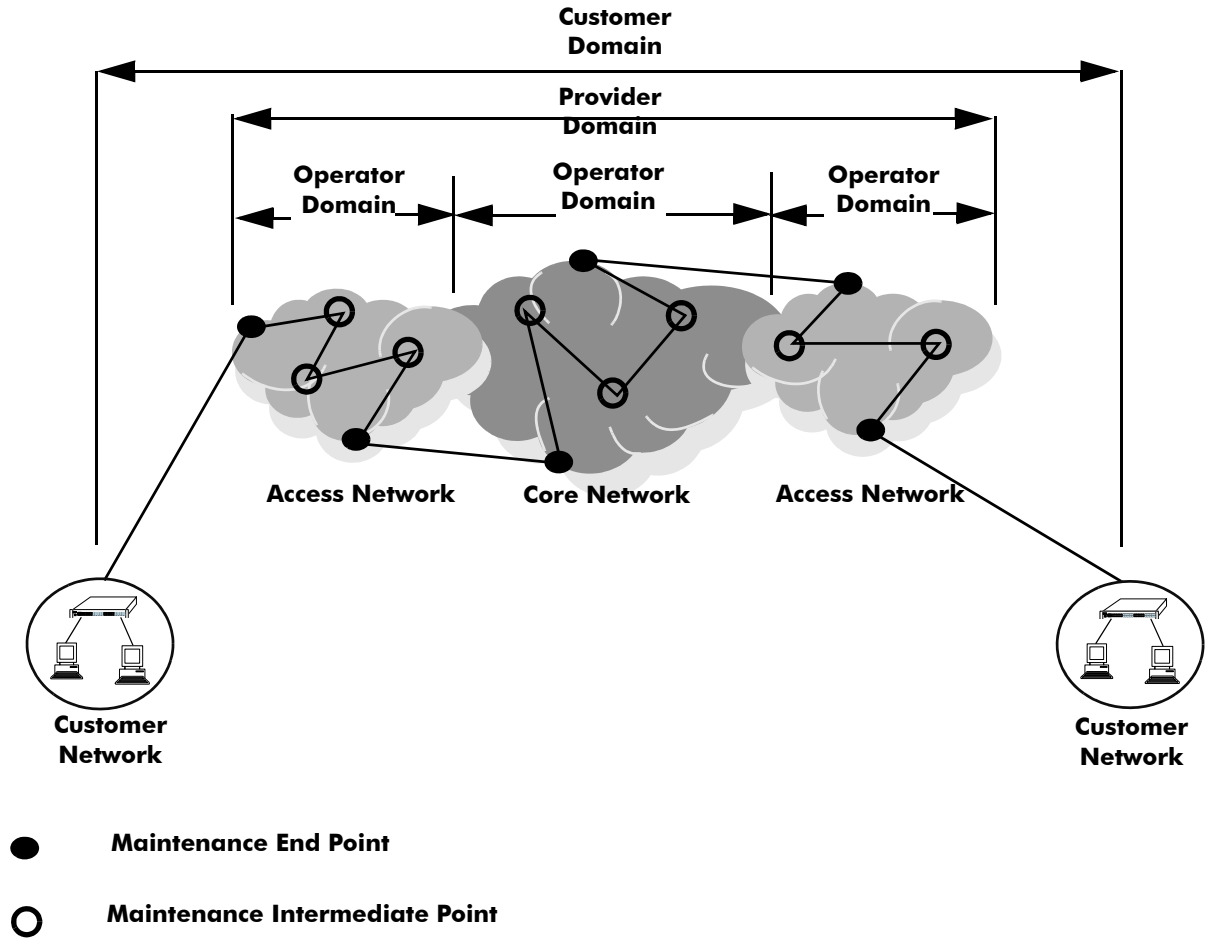


Figure 39-1 : Ethernet OAM - CFM Maintenance Domain

Fault Management

Service OAM Connectivity Fault Management consists of three types of messages that are used to help network administrators detect, verify, and isolate when a problem occurs in the network:

- **Continuity Check Messages (CCM)**—These are multicast messages exchanged periodically by MEPs to detect loss of service connectivity between MEPs. These messages are also used by MEPs and MIPs to discover other MEPs within a domain.
- **Linktrace Messages (LTM)**—These messages are transmitted by a MEP to trace the path to a destination maintenance point. The receiving maintenance point responds to LTMs with a linktrace reply (LTR). This mechanism is similar to the UDP Trace Route function. The transmission of linktrace messages is requested by an administrator.
- **Loopback Messages (LBM)**—These messages are transmitted by a MEP to a specified MIP or MEP to determine whether or not the maintenance point is reachable. The receiving maintenance point responds to LBMs with a loopback reply (LBR). This mechanism is not used to discover a path to the destination; it is similar to the Ping function. The transmission of loopback messages is requested by an administrator.

Remote Fault Propagation

Remote Fault propagation (RFP) propagates connectivity fault events into the interface that is attached to a MEP. Once the fault is detected for a MEP, the MEP's interface is shutdown. The feature is configurable on per MEP basis and is supported only for UP MEPs. It detects only loss of connectivity and remote MAC defect.

MIP CCM Database Support

Per section 19.4 of the IEEE 802.1ag 5.2 draft standard, an MHF may optionally maintain a MIP CCM database as it is not required for conformance to this standard. A MIP CCM database, if present, maintains the information received from the MEPs in the MD and can be used by the Linktrace Protocol.

This implementation of Ethernet OAM does not support the optional MIP CCM database. As per section 19.4.4 of the IEEE 802.1ag 5.2 draft standard, LTM is forwarded on the basis of the source learning filtering database. Because the MIP CCM database is not supported in this release, MIPs will not forward LTM on blocked egress ports.

Performance Monitoring

The ITU-T Y.1731 Recommendation addresses the need to monitor performance to help enforce customer service level agreements (SLAs). Frame delay (latency) and frame delay variation (jitter) are important performance objectives, especially for those applications (such as voice) that cannot function with a high level of latency or jitter.

This implementation of Service OAM supports Ethernet frame delay measurement (ETH-DM) and is compliant with Y.1731. The ETH-DM feature allows for the configuration of on-demand OAM to measure frame delay and frame delay variation between endpoints.

Frame delay measurement is performed between peer MEPs (measurements to MIPs are not done) within the same MA. Although the OmniSwitch implementation of ETH-DM is compliant with ITU-T Y.1731, delay measurement can be performed for both ITU-T Y.1731 and IEEE 802.1ag MEPs.

Any MEP can initiate or reply to an ETH-DM request, depending on the type of delay measurement requested. There are two types of delay measurements supported: one-way and two-way.

One-way ETH-DM

- A MEP sends one-way delay measurement (1DM)) frames to a peer MEP. The sending MEP inserts the transmission time into the 1DM frame at the time the frame is sent.
- When a MEP receives a 1DM frame, the MEP calculates the one-way delay as the difference between the time at which the frame was received and the transmission time indicated by the frame timestamp (receive time minus transmission time).
- One-way delay measurement statistics are gathered and stored on the receiving MEP (the MEP that receives a 1DM request).
- One-way ETH-DM requires clock synchronization between the sending and receiving MEPs. Using NTP for clock synchronization is recommended.

Two-way ETH-DM

- A MEP sends delay measurement message (DMM) frames to a peer MEP to request a two-way ETH-DM. The sending MEP inserts the transmission time into the DMM frame at the time the frame is sent.
- When a MEP receives a DMM frame, the MEP responds to the DMM with a delay message reply (DMR) frame that contains the following timestamps:
  - Timestamp copied from the DMM frame.
  - Timestamp indicating when the DMM frame was received.
  - Timestamp indicating the time at which the receiving MEP transmitted the DMR frame back to the sending MEP.
- When a MEP receives a DMR frame, the MEP compares all the DMR timestamps with the time at which the MEP received the DMR frame to calculate the two-way delay.
- The two-way delay is the difference between the time the originating MEP sent a DMM request and the time at which the originating MEP received a DMR frame minus the time taken by the responding MEP to process the DMM request.
- Two-way delay measurement statistics are gathered and stored on the originating MEP (the MEP that initiates a DMM request).
- This method *does not* require clock synchronization between the transmitting and receiving MEPs.
- Two-way ETH-DM is an on-demand OAM performance measurement. To set up continuous two-way delay measurement, see Chapter 43, “Configuring Service Assurance Agent,” for information about how to configure a SAA for continuous two-way frame delay measurement.

Frame Delay Variation

The delay variation (jitter) for both one-way and two-way ETH-DM is determined by calculating the difference between the current delay measurement value and the previous delay measurement value. If a previous delay value is not available, which is the case when a DM request is first made, then jitter is not calculated.

Interoperability with ITU-T Y.1731

This implementation of Ethernet Service OAM supports both IEEE 802.1ag and ITU-T Y.1731 for connectivity fault management (plus performance monitoring provided by ITU-T Y.1731). Although both standards are supported, the OmniSwitch implementation uses the 802.1ag terminology and hierarchy for Ethernet CFM configuration.

The following table provides a mapping of 802.1ag terms to the equivalent ITU-T Y.1731 terms:

IEEE 802.1ag v8.1	ITU-T Y.1731
Maintenance Domain (MD)	Maintenance Entity (ME)
Maintenance Association (MA)	Maintenance Entity Group (MEG)
Maintenance Endpoint (MEP)	MEG Endpoint (MEP)
Maintenance Intermediate Point (MIP)	MEG Intermediate Point (MIP)
Maintenance Domain Level	MEG Level

Support for both the IEEE and ITU-T Ethernet CFM standards allows interoperability between OmniSwitch 802.1ag and Y.1731 CFM with the following minor configuration requirements:

- The OmniSwitch MD format must be configured as “none”.
- ITU-T Y.1731 uses the “icc-based” format for a MEG, so the OmniSwitch MA format must also be configured to use the “icc-based” format.
- When the OmniSwitch MA is configured with the “icc-based” format, the MA name is automatically padded with zeros if the name specified is less than 13 characters.

The OmniSwitch CLI commands to configure an MD and MA include the “none” and “icc-based” format options. See “Configuring Ethernet OAM” on page 39-9 for more information.

Quick Steps for Configuring Ethernet OAM

The following steps provide a quick tutorial on how to configure Ethernet OAM. Each step describes a specific operation and provides the CLI command syntax for performing that operation.

1 Create an Ethernet domain using the **ethoam domain** command. For example:

```
-> ethoam domain esd.ale.com format dnsName level 1
```

2 Create an Ethernet OAM Maintenance Association using the **ethoam association** command. For example:

```
-> ethoam association ale-sales format string domain esd.ale.com vlan 10
```

3 Create an Ethernet OAM Maintenance End Point using the **ethoam endpoint** command. For example:

```
-> ethoam endpoint 100 domain esd.ale.com association ale-sales direction up
port 1/10
```

4 Administratively enable the Ethernet OAM Maintenance End Point using the **ethoam endpoint admin-state** command. For example:

```
-> ethoam endpoint 100 domain esd.ale.com association ale-sales admin-state
enable
```

5 Enable Continuity Check Messages for the Ethernet OAM Maintenance End Point using the **ethoam endpoint ccm** command. For example:

```
-> ethoam endpoint 100 domain esd.ale.com association ale-sales ccm enable
```

6 Configure the Message Handling Function (MHF) value of an Ethernet OAM Maintenance Domain using the **ethoam domain mhf** command. For example:

```
-> ethoam domain esd.ale.com mhf explicit
```

7 Configure the endpoint list for the Ethernet OAM Maintenance Association using the **ethoam association endpoint-list** command. For example:

```
-> ethoam association ale-sales domain esd.ale.com endpoint-list 100
```

8 Enable the maintenance entity to initiate transmitting loopback messages to obtain loopback replies using the **ethoam loopback** command. For example:

```
-> ethoam loopback target-endpoint 15 source-endpoint 100 domain esd.ale.com
association ale-sales
```

Configuring Ethernet OAM

This section describes how to use the OmniSwitch Command Line Interface (CLI) to configure Ethernet Service OAM on a switch. Consider the following guidelines when configuring Service OAM maintenance entities:

- Ethernet OAM is not supported on mobile, mirrored, or aggregate ports (the physical port members of an aggregate).
- Ethernet OAM is also not supported on dynamically learned VLANs.
- Implementing Ethernet OAM is supported on any full-duplex point-to-point or emulated point-to-point Ethernet link. It need not be implemented system wide.
- Management systems are important for configuring Ethernet OAM across the network. They also help to automate network monitoring and troubleshooting. Ethernet OAM can be configured in two phases: network configuration phase and service activation phase.
- The network configuration phase enables Connectivity Fault Management (CFM) on the switches. This is also the phase where Maintenance Intermediate Points (MIP) and Maintenance End Points (MEP) are identified and set up.
- Any port on a switch is referred to as a Maintenance Point (MP). An MP can be either a MEP or MIP. A MEP resides at the edge of a Maintenance Domain (MD), while a MIP is located within a MD.
- In the Service Activation phase, a new end point is created on a VLAN as a MEP. This enables the configuration of continuity-check and cross-check functionality.

Configuring a Maintenance Domain

To create a Maintenance Domain (MD), use the **ethoam domain** command, by entering **ethoam domain**, followed by the domain name, the keyword **format,** the domain name format type, the keyword **level**, and the level of the domain. For example:

```
-> ethoam domain esd.ale.com format dnsName level 5
```

Here, the MD **esd.ale.com** is created.

Note that the level must be 0-2 at operator level, 3-5 at provider level, and 6-7 at customer level when creating the level of domain.

To remove an MD, use the **no** form of this command. For example:

```
-> no ethoam domain esd.ale.com
```

Note that with this implementation of Ethernet OAM, it is only possible to delete an MD when there is no Maintenance Association, End Point, or Intermediate Point associated with the MD.

Modifying a Maintenance Domain

To modify the MHF value of an MD, use the **ethoam domain mhf** command, as shown:

```
-> ethoam domain esd.ale.com mhf explicit
```

To modify the default Ethernet OAM Maintenance Domain, use the **ethoam default-domain level** command, as shown:

```
-> ethoam default-domain vlan 100 level 4 mhf none
```

Note. The **no** form of this command restores the default Ethernet OAM Maintenance Domain value.

Configuring a Maintenance Association

To create an Ethernet OAM Maintenance Association (MA), use the **ethoam association** command. For example, to create the MA **ale-sales** in the **esd.ale.com** domain, enter:

```
-> ethoam association ale-sales format string domain esd.ale.com primary-vlan 10
```

To remove an MA, use the **no** form of this command. For example:

```
-> no ethoam association ale-sales domain esd.ale.com
```

Note that with this implementation of Ethernet OAM, it is only possible to delete an MA when there is no Maintenance End Point (MEP) or Maintenance Intermediate Point (MIP) associated with the MA.

Configuring Maintenance Association Attributes

The MIP Half Function (MHF), Continuity Check Message (CCM) interval, and MEP list are configurable attributes of a Maintenance Association.

By default, the MHF value is set to defer. To modify this value for an MA, use the **ethoam association primary vlan** command. For example:

```
-> ethoam association ale-sales domain esd.ale.com mhf default
```

By default, the CCM interval is set to 10 seconds. To modify this value for an MA, use the **ethoam association ccm-interval** command:

```
-> ethoam association ale-sales domain esd.ale.com ccm-interval interval1m
```

To modify the MEP list of an MA, use the **ethoam association endpoint-list** command, as shown:

```
-> ethoam association ale-sales domain esd.ale.com endpoint-list 100-200
```

To remove the MEP list from an Ethernet OAM Maintenance Association, enter:

```
-> no ethoam association ale-sales domain esd.ale.com endpoint-list 100-200
```

Configuring a Maintenance End Point

To create an Ethernet OAM Maintenance End Point (MEP), use the **ethoam endpoint** command. For example, to create UP MEP 100 in domain "esd.ale.com" of the "ale-sales" Maintenance Association on port 1/2 of VLAN 400, enter:

```
-> ethoam end-point 100 domain esd.ale.com association ale-sales direction up
port 1/2 primary-vlan 400
```

To remove a MEP, use the **no** form of this command. For example:

```
-> no ethoam end-point 100 domain esd.ale.com association ale-sales
```

To configure the administrative state of a MEP, use the **ethoam endpoint admin-state** command. For example:

```
-> ethoam end-point 100 domain esd.ale.com association ale-sales admin-state
enable
```

Configuring a Virtual Maintenance End Point

Virtual UP MEP is an UP MEP that is created on a 'virtual' port. This port is neither a physical port nor a logical port. This port is not connected to any switch interface. The virtual MEP will not transmit port and interface status TLVs.

The use of Virtual MEP allows to create a MEP on a virtual port thus saving the use of physical port.

To configure a virtual MEP, use the **ethoam endpoint** command. For example, to create UP MEP 100 in domain "esd.ale.com" of the "ale-sales" Maintenance Association on a virtual port of VLAN 400, enter:

```
-> ethoam end-point 100 domain esd.ale.com association ale-sales direction up
port virtual primary-vlan 400
```

Note the following when configuring the virtual MEP:

- A virtual MEP shall only be configured as an UP-MEP.
- Virtual MEP can be configured in any valid level.
- The virtual MEP is configured on a virtual port and not attached to any switch interface.
- Only one virtual MEP can be configured per switch.
- The behavior of virtual MEP will be the same as that of the MEPs created on physical ports.
- The Remote Fault Propagation feature is not supported for virtual UP MEP.

Configuring Remote Fault Propagation (RFP)

Remote Fault propagation (RFP) propagates connectivity fault events into the interface that is attached to a MEP. Once the fault is detected for a MEP, the MEP's interface is shutdown. Unlike other violation mechanisms that keep the link up when an interface is shutdown, this fault propagation mechanism will effectively shutdown the link, so that the remote end of the interface also detects a link down. RFP detects only Loss of connectivity and Remote MAC defect. To enable or disable RFP on MEP, use the command **ethoam endpoint domain association rfp**. For example:

To enable RFP on MEP,

```
-> ethoam endpoint 3 domain md1 association ma1 rfp enable
```

To disable RFP on MEP,

```
-> ethoam endpoint 3 domain md1 association ma1 rfp disable
```

Note. Remote Fault Propogation (RFP) is configurable on per MEP basis supported only for UP MEPs. RFP is not supported on virtual UP MEPs.

Configuring MEP Attributes

To configure the MEP to generate Continuity Check Messages (CCM), use the **ethoam endpoint ccm** command. For example:

```
-> ethoam end-point 100 domain esd.ale.com association ale-sales ccm enable
```

To configure the priority values for Continuity Check Messages and Linktrace Messages transmitted by a MEP, use the **ethoam endpoint priority** command. For example:

```
-> ethoam end-point 100 domain esd.ale.com association ale-sales priority 6
```

To configure the lowest priority fault alarm for the lowest priority defect for a MEP, use the **ethoam endpoint lowest-priority-defect** command. For example:

```
-> ethoam end-point 100 domain esd.ale.com association ale-sales lowest-
priority-defect all-defect
```

Configuring Loopback

To initiate transmitting Loopback messages (LBMs) and obtaining Loopback replies (LBRs), use the **ethoam loopback** command. For example:

```
-> ethoam loopback target-endpoint 10 source-endpoint 20 domain MD association
MA number 3
Reply from 00:0E:B1:6B:43:89: bytes=64 seq=0 time=100ms
Reply form 00:0E:B1:6B:43:89: bytes=64 seq=0 time=112ms
Request timed out.
----00:E0:B1:6B:43:89 ETH-LB Statistics----
3 packets transmitted, 2 packets received, 33% packet loss
round-trip (ms)  min/avg/max = 100/106/112
```

Configuring Linktrace

To initiate transmitting Linktrace messages (LTMs) and detecting Linktrace replies (LTR), use the **ethoam linktrace** command. For example:

```
-> ethoam linktrace 10:aa:ac:12:12:ad end-point 4 domain esd.ale.com association
ale_sales flag fdbonly hop-count 32
```

Configuring the Fault Alarm Time

The Fault Alarm time is the period of time during which one or more defects should be detected before the Fault Alarm is issued. By default, this timer is set to 250 centiseconds. To change the Fault Alarm time, use the **ethoam fault-alarm-time** command. For example:

```
-> ethoam fault-alarm-time 500 end-point 100 domain esd.ale.com association
ale_sales
```

Configuring the Fault Reset Time

The Fault Reset time is the time interval in which Fault Alarm is re-enabled to process the faults. By default, this timer value is set to 1000 centiseconds. To change the Fault Reset time, use the **ethoam fault-reset-time** command. For example:

```
-> ethoam fault-reset-time 250 end-point 100 domain esd.ale.com association
ale_sales
```

Configuring Ethernet Frame Delay Measurement

Ethernet frame delay measurement (ETH-DM) is an on-demand OAM function used to measure frame delay (latency) and delay variation (jitter) between MEPs. There are two types of ETH-DM supported: one-way and two-way.

One-Way ETH-DM

The **ethoam one-way-delay** command is used to configure a one-way ETH-DM (1DM) to monitor performance between two MEPs. For example, the following command is used to initiate the transmission of 1DM frames to a target MEP:

```
-> ethoam one-way-delay target-endpoint 10 source-endpoint 12 domain MD1
association MA1 vlan-priority 4
```

This command initiates the sending of 1DM frames from MEP 12 to MEP 10, which does not reply to frames received from MEP 12. The latency and jitter statistics are gathered and stored on the receiving MEP, which is MEP 10 in this example.

An option to specify a target MAC address, instead of a MEP ID, is also supported. For example:

```
-> ethoam one-way-delay target-macaddress 00:e0:b1:6a:52:4c source-endpoint 12
domain MD association MA vlan-priority 4
```

One-way delay measurement statistics are gathered and stored on the receiving MEP (the MEP that receives a 1DM request).

Note. One-way ETH-DM requires clock synchronization between the sending and receiving MEPs. Using NTP for clock synchronization is recommended.

Two-Way ETH-DM

The **ethoam two-way-delay** command is used to configure a two-way ETH-DM to monitor roundtrip performance between two MEPs. For example, the following command is used to initiate the transmission of delay measurement message (DMM) frames to a target MEP:

```
-> ethoam two-way-delay target-endpoint 10 source-endpoint 12 domain MD
association MA vlan-priority 4
Reply from 00:0E:B1:6B:43:89 delay=2584us jitter=282us
```

This command initiates the sending of DMM frames from MEP 12 to MEP 10. However, with two-way delay measurement, the receiving MEP replies with delay message response (DMR) frames to the sending MEP. In this example, MEP 10 sends DMR frames back to MEP 12.

An option to specify a target MAC address, instead of a MEP ID, is also supported. For example:

```
-> ethoam two-way-delay target-macaddress 00:e0:b1:6a:52:4c source-endpoint 12
domain MD association MA vlan-priority 4
Reply form 00:E0:B1:6A:52:4C: delay=2584us jitter=282us
```

Note the following when configuring two-way ETH-DM:

- Two-way delay measurement statistics are gathered and stored on the originating MEP (the MEP that initiates a DMM request).
- This method *does not* require clock synchronization between the transmitting and receiving MEPs.
- Two-way ETH-DM is an on-demand OAM performance measurement. To schedule continuous two-way delay measurement, see Chapter 43, “Configuring Service Assurance Agent,” for more information.

CVLANs for EVC MEPs

MEPs on per customer VLAN (CVLAN) in a SVLAN allows supporting connectivity and fault management on per CVLAN basis. MEP can be configured on the UNI-N of the provider bridge. The CVLAN based MEP will assist the service provider to instantiate an MEP instance for the customer VLAN on the UNI-N port to perform OAM action for individual CVLAN traffic bound to the EVC.

The following sections provide the details about the CVLAN configuration for the EVC MEPs.

CVLAN Configuration Overview

Ethernet-service must be configured prior to configuring MEP on CVLAN.

The CVLAN for the EVC MEP must be configured only on UP MEPs and User-Network (UNI) port. The CVLAN MEPs associated with the EVC MEG will generate and receive double tagged frames. Maximum of two tags is processed.

The CVLAN associated to the MEP must be part of the allowed CVLAN list configured in the MA for which the MEP belongs.

Creating MEP on CVLAN

MEPs can be created on per CVLAN in a SVLAN. To create a MEP on a CVLAN, use the command **ethoam endpoint** command. For example:

```
-> ethoam endpoint 10 domain MD association MA direction up port 1/1/1 cvlan 20
```

The direction of the MEP must be UP and the port should be a UNI port using Ethernet service, for which the CVLAN is configured.

Note. The CVLAN ID must be part of the allowed CVLAN list before being configured.

Configuring the Allowed CVLAN List

The CVLAN ID before being associated to a MEP for monitoring must be configured in the allowed CVLAN list. To configure the allowed CVLAN list, use the **ethoam association allowed-cvlan-list** command. For example:

```
-> ethoam association MA domain MD allowed-cvlan-list 10-15
```

Note. The CVLANs configured in the allowed CVLAN list must be associated with the SVLAN of the MA.

Configuring the CVLAN (ctag) Priority

Configure the priority value to be included in the inner tag of the CFM frames originating from CVLAN MEPs.

To configure the priority, use the **ethoam endpoint ctag-priority** command. For example:

```
-> ethoam endpoint 1 domain md1 association ma1 ctag-priority 6
```

If the priority value is not configured, by default, the outer-tag priority is considered as the inner-tag priority. The priority value range is 0 to 7.

Viewing the CVLAN Configurations

The following commands displays the CVLAN related information:

show ethoam domain association	Displays the CVLANs configured for the MEPs in the MA.
show ethoam domain association end-point	Displays the CVLAN configured for the MEP.
show ethoam domain	Displays the allowed CVLAN list configured for the MEP.
show ethoam cfmstack	Displays the contents of CFM Stack Managed Object, which determines the relationships among MEPs and MIPs on a specific bridge port.

Verifying the Ethernet OAM Configuration

To display information about Ethernet OAM on the switch, use the show commands listed below:

show ethoam	Displays the information of all the Management Domains configured on the switch.
show ethoam domain	Displays the information of a specific Management Domain configured on the switch.
show ethoam domain association	Displays the information of a specific MA in a Management Domain configured on the switch.
show ethoam domain association end-point	Displays the information of a specific MEP in a Management Domain configured on the switch.
show ethoam remote-endpoint domain	Displays the information of all remote MEPs learned as a part of the CCM message exchange.
show ethoam default-domain configuration	Displays all the default MD information for all the VLANs or a specific VLAN.
show ethoam default-domain configuration	Displays the values of scalar Default-MD objects
show ethoam vlan	Displays the vlan association for a specified VLAN-ID
show ethoam cfmstack	Displays the contents of CFM Stack Managed Object, which determines the relationships among MEPs and MIPs on a specific switch port.
show ethoam linktrace-reply	Displays the content of the Linktrace reply (LTR) returned by a previously transmitted LTM. This command displays the LTR based on the transaction identifier or sequence number of the LTM for which the LTR is to be displayed
show ethoam linktrace-tran-id	Displays the transaction identifiers returned by previously generated LTMs from a specified MEP.
show ethoam statistics	Displays the Ethernet OAM statistics of all the Management Domains configured on the switch. Also, displays the statistics of all the MAs and matching MEPs for all the MDs.
show ethoam config-error	Displays the configuration error for a specified VLAN, port or linkagg.

40 Configuring EFM (LINK OAM)

Ethernet in the First Mile (EFM), also known as LINK OAM, is a collection of protocols specified in IEEE 802.3ah, defining Ethernet in the access networks that connects subscribers to their immediate service provider. EFM, EFM-OAM and LINK OAM refers to IEEE 802.3ah standard.

LINK OAM (Operation, Administration, and Maintenance) is a tool monitoring Layer-2 link status by sending OAM protocol data units (OAMPDUs) between networked devices on the first mile. The first mile network refers to the connection between the subscriber and the public carrier network. LINK OAM is mainly used to address common link-related issues on the first mile. It helps network administrators manage their networks effectively.

By enabling LINK OAM on two devices connected by a point-to-point connection, network administrators can monitor the status of the link, detect faults in network segments, and probe link errors by using loopback testing.

In This Chapter

This chapter describes the LINK OAM feature and how to configure it through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *CLI Reference Guide*. This chapter provides an overview of LINK OAM and includes the following information:

- "LINK OAM Defaults" on page 40-2
- "Quick Steps for Configuring LINK OAM" on page 40-3
- "Interaction With Other Features" on page 40-6
- "Configuring Link Monitoring" on page 40-8
- "Configuring LINK OAM" on page 40-7
- "Verifying the LINK OAM Configuration" on page 40-10

LINK OAM Defaults

The following table shows LINK OAM default values.

Parameter Description	Command	Default Value/Comments
Multiple PDU count assigned for event notifications.	**efm-oam multiple-pdu-count**	3
Maximum time period for which a LINK OAM port shall wait for a hello message from its peer before resetting a discovery session.	**efm-oam port keepalive-interval**	5 seconds
Time interval (in seconds) by which the information OAMPDUs are transmitted out of an LINK OAM enabled port.	**efm-oam port hello-interval**	1 second
Propagate local event notifications to the remote peer.	**efm-oam port propagate-events**	*critical event* - enabled *dying-gasp event* - enabled.
The threshold, window frame values and notify status for errored frame period events.	**efm-oam errored-frame-period**	*threshold_symbols* - 1 frame error *window_frames* - Depends on port types. *notify status* - enable
The threshold, window, and notify status for errored frame events.	**efm-oam errored-frame**	*threshold_symbols* - 1 frame error *window_seconds* - 1 second *notify status* - enable
The threshold, window and notify-status for errored-frame-seconds-summary on a port.	**efm-oam errored-frame-seconds-summary**	*threshold_symbols* - 1 errored frame second *window_seconds* - 60 seconds. *notify status* - enable
The number of frames sent by the current LINK OAM port to the MAC address of the remote port, the delay between the frames sent, and whether or not to start the ping operation.	**efm-oam port l1-ping**	*number* - 5 frames *milliseconds* - 1000

Quick Steps for Configuring LINK OAM

The following steps provide a quick tutorial on how to configure LINK OAM. Each step describes a specific operation and provides the CLI command syntax for performing that operation.

1 Enable LINK OAM globally on the switch by using the **efm-oam admin-state** command. For example:

```
-> efm-oam admin-state enable
```

2 Enable LINK OAM protocol for a specific port using the **efm-oam port admin-state** command. For example

```
-> efm-oam port 1/1/1 admin-state enable
```

3 Configure the LINK AOM port to active mode by using the **efm-oam port mode** command. For example:

```
-> efm-oam port 1/1/1 mode active
```

Note. The above step is optional. By default, LINK OAM mode is active on all ports.

---

4 Configure the timeout interval (keep-alive) for the dynamically learned neighboring devices on the port by using the **efm-oam port keepalive-interval** command. For example:

```
-> efm-oam port 1/1/1 keepalive-interval 10
```

5 Configure the time interval by which the information OAMPDUs has to be transmitted out of an LINK OAM enabled port by using the **efm-oam port hello-interval** command. For example:

```
-> efm-oam port 1/1/1 hello-interval 5
```

6 Activate remote loop back processing on the port by using the **efm-oam port remote-loopback** command. For example:

```
-> efm-oam port 1/1/1 remote-loopback process
```

7 Activate propagation of critical events and dying gasp events on the port by using the **efm-oam port propagate-events** command. For example:

```
-> efm-oam port 1/1/1 propagate-events critical-event enable
```

```
-> efm-oam port 1/1/1 propagate-events dying-gasp enable
```

Note. The above step is optional. By default, propagation of critical events and dying gasp is enabled on the port.

---

8 Configure the threshold, window frame values and notify status for errored frame period events on the port by using the **efm-oam errored-frame-period** command. For example:

```
-> efm-oam port 1/1/1 errored-frame-period window 3000000 threshold 1 notify
enable
```

9 Configure the threshold, window, and notify status for errored frame events on the port by using the **efm-oam errored-frame** command. For example:

```
-> efm-oam port 1/1/1 errored-frame window 32 threshold 10 notify enable
```

10 Configure the threshold, window and notify-status for errored-frame-seconds-summary on the port by using the **efm-oam errored-frame-seconds-summary** command. For example:

```
-> efm-oam port 1/1/1 errored-frame-seconds-summary window 700 threshold 1
notify enable
```

LINK OAM Overview

IEEE standard 802.3ah provides support for LINK OAM. The Clause 57 of std. 802.3ah defines the Operations, Administration, and Maintenance (OAM) sub layer, which provides mechanisms useful for monitoring link operation such as remote fault indication and remote loopback control. LINK OAM provides network operators the ability to monitor the health of the network and quickly determine the location of failing links or fault conditions.

LINK OAM provides an OAMPDU-based mechanism to notify the remote DTE when one direction of a link is non-operational and therefore data transmission is disabled. The ability to operate a link in a unidirectional mode for diagnostic purposes supports the maintenance objective of failure detection and notification.

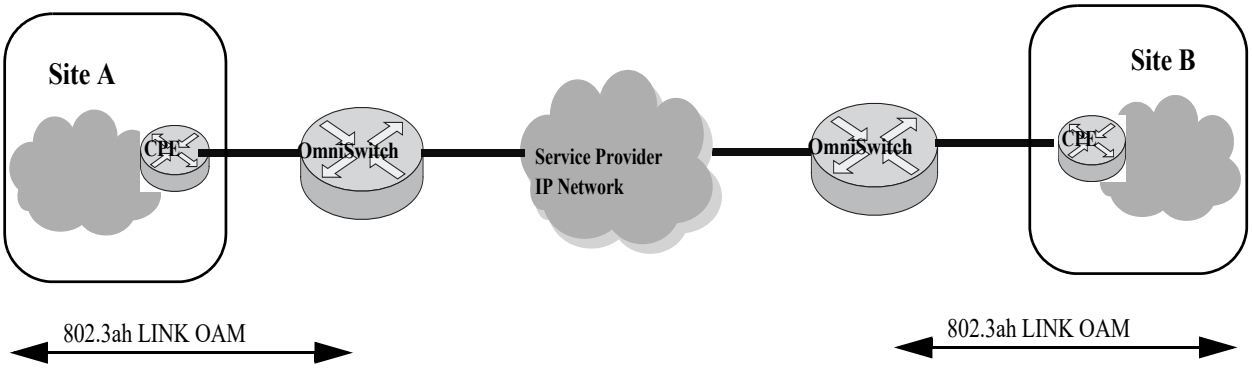


Figure 40-1 : Example LINK OAM

OAM information is conveyed in slow protocol frames called OAM Protocol Data Units (OAMPDUs). OAMPDUs contain the appropriate control and status information used to monitor, test and troubleshoot OAM-enabled links. OAMPDUs traverse a single link, being passed between peer OAM nodes, and as such, are not forwarded by MAC clients (e.g., bridges or switches). OAM does not include functions such as station management, bandwidth allocation or provisioning functions.

The mandatory LINK OAM functions include discovery operations (determining if the other end of the link is OAM capable and what OAM functions it supports), state machine implementation and some critical event flows. OAM remote loopback can be used for fault localization and link performance testing.

The features of the LINK OAM protocol discussed in this section are:

- "Discovery" on page 40-5
- "Link Monitoring" on page 40-5
- "Remote Fault detection" on page 40-5
- "Remote Loopback Testing" on page 40-6

Discovery

Discovery is the first phase of the IEEE 802.3ah OAM protocol. During discovery, information about LINK OAM node's capabilities, configuration, and identity are exchanged in the form of OAM protocol data units (OAMPDUs).

The interconnected LINK OAM nodes notify the peer of their OAM configuration information and the OAM capabilities of the local nodes by exchanging Information OAMPDUs and determine whether LINK OAM connections can be established. A LINK OAM connection between two nodes is established only when the settings concerning Loopback, link detecting, and link event of the both sides match.

Note. LINK OAM requires that frames be exchanged with a minimum frequency to maintain the relationship(keep-alive). If no OAMPDUs are received in a 5 second window, the OAM peering relationship is lost and must be restored to perform OAM functions. Use **efm-oam port keepalive-interval** command to configure the keepalive time interval.

Link Monitoring

Error detection in an Ethernet network is difficult, especially when the physical connection in the network is not disconnected but network performance is degrading gradually. Link monitoring is used to detect and indicate link faults in various environments. Link monitoring uses the Event Notification OAMPDU, and sends events to the remote OAM node when there is a disorder detected on the link. The error events defined are:

Errored frame event - An errored frame event occurs when the number of detected error frames over a specific interval exceeds the predefined threshold.

Errored frame period event - An errored frame period event occurs if the number of frame errors in specific number of received frames exceeds the predefined threshold.

Errored frame seconds event - When the number of error frame seconds detected on a port over a detection interval reaches the error threshold, an errored frame seconds event occurs.

For configuring errored frame, errored frame period, and errored frame seconds events on a port, see "Configuring Link Monitoring" on page 40-8

Remote Fault detection

In a network where traffic is interrupted due to device failures or unavailability, the flag field defined in OAMPDUs allows a LINK OAM enabled node to send severe error conditions to its peer. The severe error conditions that can be identified are:

Dying Gasp - This flag is raised when a node is about to reset, reboot, or otherwise go to an operationally down state. (An unexpected fault, such as power failure has occurred.)

Critical Event - This flag indicates a severe error condition that does not result in a complete reset or reboot by the peer node. (An undetermined critical event happened.)

One of the most critical problems in an access network for carriers is differentiating between a simple power failure at the customer premise and an equipment or facility failure. Dying gasp provides this information by having a node indicate to the network that it is having a power failure. More details on the failure may be included in additional event information conveyed in the frame.

For setting up the notification of critical events on a port, see "Enabling and Disabling Propagation of Events" on page 40-8

Remote Loopback Testing

Remote loopback, which is often used to troubleshoot networks, allows one node to put the other node into a state whereby all inbound traffic is immediately reflected back onto the link. Remote loopback is most useful as a diagnostic tool, where it can be used to isolate problem segments in a large network.
By performing remote loopback tests periodically, network administrators can detect network faults in time and also isolate the network segments where errors have occurred.

Remote loopback testing in networks can be done only after the LINK OAM connection is established. With remote loopback enabled, the LINK OAM node operating in active LINK OAM mode issues remote loopback requests and the peer responds to them. If the peer operates in the loopback mode, it returns all the PDUs except Ethernet OAMPDUs to the senders along the original paths.

For enabling or disabling remote loopback process on a port, see "Enabling and Disabling Remote loopback" on page 40-9

Interaction With Other Features

This section contains important information about how other OmniSwitch features interact with LINK OAM. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

Link Aggregate

LINK OAM does not work on the logical link aggregate port. But, it can run on the individual aggregable (physical) port.

Connectivity Fault Management

Connectivity Fault Management (IEEE 802.1ag) covers the scope of Ethernet service over any path, whether a single link or end-to-end, enabling service providers to fully monitor Ethernet service regardless of the layers supporting the service, the network path, or the various network operators involved. It divides a network into maintenance domains in the form of hierarchy levels, which are then allocated to users, service providers and operators.

Connectivity Fault Management (CFM) assigns maintenance end points (MEPs) to the edges of each domain and maintenance intermediate points (MIPs) to ports within domains. This helps to define the relationships between all entities from a maintenance perspective, to allow each entity to monitor the layers under its responsibility and localize the errors easily.

ERP

LINK OAM is supported in Ethernet Ring Protection (ERP) switching mechanism. ERP (ITU-T G.8032/Y.1344) is a self-configuring algorithm that maintains a loop-free topology while providing data path redundancy and network scalability. ERP provides fast recovery times for Ethernet ring topologies by utilizing traditional Ethernet MAC and bridge functions.

Configuring LINK OAM

This section describes how to use Alcatel-Lucent's Command Line Interface (CLI) commands to configure LINK OAM on a switch.

Enabling and Disabling LINK OAM

The **efm-oam admin-state** is used to enable LINK OAM globally. By default, LINK OAM is disabled on the switch. The **efm-oam port admin-state** command can be used to enable or disable the LINK OAM on a specific port or a range of ports on a switch. When enabled, the port can be set to receive, transmit, or both transmit and receive OAMPDUs.

To enable LINK OAM globally on a range of ports, use the **efm-oam** command, as shown:

```
-> efm-oam port 2/1/1-10 status enable
```

To disable LINK OAM globally on a range of ports, use the **disable** form of the command, as shown:

```
-> efm-oam port 2/1/1-10 status disable
```

To enable LINK OAM mode to active, use the **port mode** command, as shown:

```
-> efm-oam port 2/1/1-10 mode active
```

By default, LINK OAM port mode is active on all the ports.

Setting the Transmit Delay

LINK OAM requires that frames be exchanged with a minimum frequency to maintain the relationship (keep-alive). If no OAMPDUs are received in a specific time interval window, the OAM peering relationship is lost and must be restored to perform OAM functions.

Use **efm-oam port keepalive-interval** command to configure the keepalive time interval.

```
-> efm-oam port 2/1/1-10 keepalive-interval 10
```

To configure the time interval by which the information OAMPDUs has to be transmitted out of an LINK OAM enabled port, use the **efm-oam port hello-interval** command.

```
-> efm-oam port 2/1/1-10 hello-interval 10
```

Note. By default, the keep-alive interval value is 5 seconds and the hello-interval value is set to 1 second.

Enabling and Disabling Propagation of Events

In a network where traffic is interrupted due to device failures or unavailability, the flag field defined in OAMPDUs allows a LINK OAM enabled node to send severe error conditions to its peer. See "Remote Fault detection" on page 40-5 for more information on error conditions.

The ports can be enabled to report severe error conditions like critical events and dying gasp events by using the **efm-oam port propagate-events** command.

```
-> efm-oam port 2/1/1-10 propagate-events critical-event enable

-> efm-oam port 2/1/1-10 propagate-events dying-gasp enable
```

Note. The above commands are optional. By default, propagation of critical events and dying gasp is enabled on the port.

Configuring Link Monitoring

Link monitoring is used to detect and indicate link faults in various environments. Link monitoring uses the Event Notification OAMPDU, and sends events to the remote OAM node when there is a disorder detected on the link. For more information on error events, see "Link Monitoring" on page 40-5

Enabling and Disabling Errored frame period

Configure the threshold, window frame values and notify status for errored frame period events on the port by using the **efm-oam errored-frame-period** command.

```
-> efm-oam port 2/1/1-10 errored-frame-period window 3000000 threshold 1 notify
enable
```

To disable notification of errored frame period events, use the following command.

```
-> efm-oam port 2/1/1-10 errored-frame-period notify disable
```

Enabling and Disabling Errored frame

Configure the threshold, window, and notify status for errored frame events on the port by using the **efm-oam errored-frame** command.

```
-> efm-oam port 2/1/1-10 errored-frame window 32 threshold 10 notify enable
```

To disable notification of errored frame events, use the following command.

```
-> efm-oam port 2/1/1-10 errored-frame notify disable
```

Enabling and Disabling Errored frame seconds summary

Configure the threshold, window and notify-status for errored-frame-seconds-summary on the port by using the **efm-oam errored-frame-seconds-summary** command.

```
-> efm-oam port 2/1/1-10 errored-frame-seconds-summary window 700 threshold 1
notify enable
```

To disable notification of errored frame events, use the following command.

```
-> efm-oam port 2/1/1-10 errored-frame-seconds-summary notify disable
```

Configuring LINK OAM Loopback

Remote loopback is most useful as a diagnostic tool, where it can be used to isolate problem segments in a large network. See "Remote Loopback Testing" on page 40-6 for more information.

Enabling and Disabling Remote loopback

LINK OAM loopback testing can be performed only after the LINK OAM connection is established and the hosts are operating in active LINK OAM mode.

When the remote-loopback is in **process** mode, the session started by peer LINK OAM client is processed by local LINK OAM port. As a result, remote port is in remote-loopback state and the local port is in local-loopback state.

Activate remote loop back processing on the port by using the **remote-loopback** command.

```
-> efm-oam port 2/1/1-10 remote-loopback process
```

When the remote-loopback is in **ignore** mode, the session started by peer LINK OAM is not processed by the local port.

For remote loop back processing to be ignored on the port, use the following command.

```
-> efm-oam port 2/1/1-10 remote-loopback ignore
```

After configuring the port to process remote loopback, the port has to be initiated for loopback session to start.

```
-> efm-oam port 2/1/1 remote-loopback start
```

The above command initiates the loopback control PDU towards the peer port to start. To stop the remote-loopback sessionl, use the following command.

```
-> efm-oam port 2/1/1 remote-loopback stop
```

To configure the number of frames to be sent by the current LINK OAM port to the remote port's MAC address (l1 ping) and the delay between each consecutive sent frames and to start the ping operation, use the following command.

```
-> efm-oam port 1/1/20 l1-ping num-frames 12 delay 500 start
```

Note. By default, the number of frames value is 5 frames and the delay is set to 1000 milliseconds.

Verifying the LINK OAM Configuration

To display information about LINK OAM on the switch, use the show commands listed below:

show efm-oam configuration	Displays the global LINK OAM configuration.
show efm-oam port	Displays the status of LINK OAM on all the ports in the system, along with other relevant information such as OAM mode, operational status and loopback status of the port.
show efm-oam port detail	Displays the LINK OAM configuration and other related parameters for a port.
show efm-oam port statistics	Displays the LINK OAM statistics on a port, or a range of ports or on all ports.
show efm-oam port remote detail	Displays the LINK OAM configuration and details of the related parameters of the remote port.
show efm-oam port history	Displays the log of events that have occurred on a port. Use this command to display specific event logs on a port.
show efm-oam port l1-ping detail	Displays the frames lost during a loopback session.

41 Configuring CPE Test Head

The Customer Provider Edge (CPE) Test Head traffic generator and analyzer is a Test-OAM (Operation, Administration and Maintenance) tool used in the Metro Ethernet Network to validate the customer Service Level Agreements (SLA). This functionality allows the operator to validate the Metro Ethernet Network between customer end points, which is critical when provisioning or troubleshooting network services.

This implementation of CPE Test Head supports Unidirectional and Bidirectional, ingress tests. Traffic is generated at the UNI port as if the traffic was generated from a test head connected to the UNI port.This validates the actual customer SLA by subjecting the test traffic to the ingress QoS defined at the UNI port (Ethernet SAP profile or QoS policy rules for priority and bandwidth control) and the egress QoS defined at the egress NNI port and carrier network.

In unidirectional test, the test traffic is unidirectional. The traffic analysis is performed by the analyzer switch.

In bidirectional test, the test traffic is bidirectional. The traffic analysis is performed by the generator switch. The test traffic is sent to the generator switch using the hardware loopback function on the analyzer switch (Loopback switch).

The feature provides single-stream and multi-stream test capability.

The CPE test is non-disruptive to traffic running on other UNI ports that are associated with the same SAP profile as the test UNI port. All UNI ports, including CPE test ports, are subject to any SAP profile or QoS configuration associated with the port. This is important to consider when analyzing test results.

In This Chapter

This chapter describes the CPE Test Head feature, CPE Test Group feature, and how to configure it through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. This includes the following information:

- "Quick Steps for Configuring CPE Test Head" on page 41-3
- "CPE Test Head Overview" on page 41-5
- "CPE Test Head Configuration Overview" on page 41-6
- "Configuring a CPE Test Profile" on page 41-7
- "Configuring the L2 SAA Test" on page 41-9
- "Running a CPE Test" on page 41-10
- "Verifying the CPE Test Configuration and Results" on page 41-11
- "Configuring CPE Test Group" on page 41-13
- "CPE Test Head Advanced Configuration" on page 41-25
- "Sample Test Configurations" on page 41-27

Quick Steps for Configuring CPE Test Head

The following steps provide a quick tutorial on how to configure a CPE test profile and run a CPE test. Each step describes a specific operation and provides the CLI command syntax that is used to perform that operation.

Configure the Test Profile

The CPE test profile is configured on both the generator and analyzer switch. Steps 1 through 5 configure profile parameters common to both the generator and analyzer switch. Steps 6 through 8 configure profile parameters required only for the generator.

1 Configure the name for the CPE test, use the **test-oam** command. For example:

```
-> test-oam Test1 descr First-test
```

2 Configure the source and destination end point for the test, use the **test-oam direction** command. For example:

```
-> test-oam Test1 src-endpoint SW1
```

```
-> test-oam Test1 dst-endpoint SW2
```

3 Configure the source MAC address, destination MAC address and the SVLAN for the test frame using the **test-oam vlan test-frame** command. For example:

```
-> test-oam Test1 vlan 100 test-frame src-mac 00:00:00:00:00:01 dst-mac
00:00:00:00:00:02
```

4 Configure the test direction using the **test-oam direction** command. For example:

```
-> test-oam Test1 direction unidirectional
```

5 Configure the type of role the switch will perform using the **test-oam role** command. For example:

```
-> test-oam Test1 role generator
```

6 Configure the test port on the switch using the **test-oam port** command. For example:

```
-> test-oam Test1 port 1/1/1
```

7 Configure the test packet parameters using the **test-oam frame** command. For example:

To configure a Layer 2 test frame, specify a hexadecimal Ether type value.

```
-> test-oam Test1 frame vlan-tag 1 priority 2 drop-eligible false ether-type
0x0100 data-pattern 0x0010
```

To configure a Layer 3 test frame, specify **ipv4** as the Ether type value.

```
-> test-oam Test1 frame vlan-tag 1 priority 2 drop-eligible false ether-type
ipv4 src-ip 1.1.1.1 dst-ip 2.2.2.2 ttl 4 tos 0x01 protocol udp src-port 2000
dst-port 3000 data-pattern 0x0010
```

8 Configure the test duration, rate and packet-size using the **test-oam duration rate packet-size** command. For example:

```
-> test-oam Test1 duration 10 rate 8kbps packet-size 64
```

Running the Test

1 Start the test on the analyzer switch first and then on the generator switch using the **start** option of the **test-oam l2-saa** command. For example:

```
-> test-oam Test1 start
```

For bidirectional test use the f**etch-remote-stats** parameter with the **test-oam l2-saa** command. For example:

```
-> test-oam Test1 start fetch-remote-stats
```

When the test runs the amount of time specified for the test duration, the test automatically stops.

2 To stop an active test from running, use the **stop** form of the **test-oam l2-saa** command.
For example:

```
-> test-oam Test1 stop
```

Note. Verify the test configuration and status with the **test-oam statistics flash-logging** command. For example:

```
-> show test-oam tests
Total Test-Ids: 1
Test-Id Port       Src-Mac        Dst-Mac         Vlan     Direction     Status         Remote-Sys-Mac
-------+-----+----------------+----------------+-----+--------------+-----------+---------------------
Test1   none  00:00:00:00:00:00  00:00:00:00:00:00 none  unidirectional  not-started  00:00:00:00:00:00
```

To verify test results, use the **show test-oam statistics** command. For example:

```
-> show test-oam Test1 statistics
Test-Id   TX-Ingress   TX-Egress    RX-Ingress      Remote-Stats    Throughput(Mbs)
--------+------------+------------+---------------+----------------+--------------
Test1     19017        19017         19017             19017         9.98
```

To clear test statistics, use the **show test-oam saa statistics** command. For example:

```
-> clear test-oam Test1 statistics
```

This clears all the statistics related to “Test1”.

```
-> clear test-oam statistics
```

This will clear the statistics for all the tests.

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about these commands.

CPE Test Head Overview

The OmniSwitch CPE Test Head feature provides a remote test generator and analyzer/loopback capability for testing and validating the customer Ethernet service domain from end-to-end. This allows the service provider to perform the following tasks without the need for an external test head device:

- Generate specific flow-based traffic across the customer's Ethernet Virtual Circuit (EVC) to help identify flow-based issues.
- Identify the impact of QoS settings (SAP profile or QoS policies) on the overall traffic.
- Confirm throughput across the provider network.
- Debug flow-specific traffic forwarding across the provider network.
- Analyze the behavior of various user-defined traffic patterns across the provider network.
- Perform the handover testing after initial deployment.
- Perform on-demand testing and results monitoring using a central entity.

The OmniSwitch implementation of CPE Test Head supports the ability to run unidirectional, ingress tests.Test setup involves configuring one CPE switch as the generator and a remote switch as the analyzer/loopback.

The following diagram shows an example of an OmniSwitch CPE Test Head configuration:

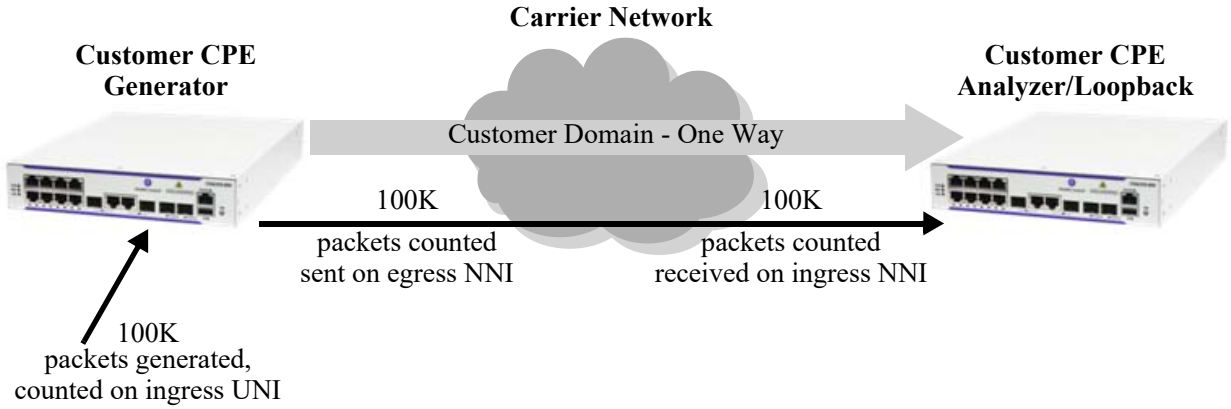


Figure 41-1 : CPE Test Head Example - Unidirectional, Ingress Test

In this example:

1 The CPE test is started first on the analyzer/loopback switch and then on the generator switch. The analyzer/loopback switch sends packets to the generator switch to learn the source.

2 A configurable amount of traffic is generated and counted on the ingress UNI port of the generator switch, as if the traffic was generated from a test head connected to the UNI port. This subjects the test traffic to the ingress UNI SAP profile policies.

3 Traffic is counted and sent out on the SAP NNI port. This subjects the test frames to the egress NNI QoS policies.

4 Test frames are forwarded through the provider network over the customer EVC to the ingress NNI on the analyzer switch, where the packets are received and counted. Note that test frames are dropped after they are counted.

5 CPE Test Head CLI **show** commands are used on the generator and analyzer switches to display and verify test statistics, such as packets transmitted and received.

Note. The CPE test is non-disruptive to traffic running on other UNI ports that are associated with the same SAP profile as the test UNI port. All UNI and NNI ports, including CPE test ports, are subject to any SAP profile or QoS configuration associated with the SAP profile. This is important to consider when analyzing test results.

CPE Test Head Configuration Overview

CPE Test Head configuration is done using a test profile to define test attributes. Configuring a test profile is required on both the generator and analyzer/loopback switch. Not all test profile information is required for both switches. For example, the profile on the generator switch must contain a port number to identify the UNI port on which the test will run, but a port number is not required for the analyzer profile.

The following table provides a list of test profile parameters and identifies if the parameter is required on the generator, analyzer, or both. Also included is the CLI command used to configure the parameter.

Test Profile Parameters	Generator Switch	Analyzer/ Loopback Switch	CLI Command
Profile name	Yes	Yes	**test-oam**
Source and destination endpoints	Yes	Yes	**test-oam direction**
Test frame source and destination MAC addresses	Yes	Yes	**test-oam vlan test-frame**
Service VLAN	Yes	Yes	**test-oam vlan test-frame**
Test role (generator or analyzer or loopback)	Yes	Yes	**test-oam role**
UNI port for test packet generation	Yes	No	**test-oam port**
Test frame parameters, such as VLAN tag, priority, and frame type	Yes	Yes	**test-oam frame**
Test duration, rate, and packet size	Yes	No	**test-oam duration rate packet-size**
Remote Sys MAC	Yes	No	**test-oam remote-sys-mac**

Configuration Guidelines

Consider the following guidelines when configuring the OmniSwitch CPE Test Head:

- Make sure the same test profile name (test ID) is used on the generator and analyzer/loopback switch.
- A switch can only perform one role (generator or analyzer or loopback) for a specific test.
- Only one test can be active for the switch at any given time.
- Up to 32 test profiles are allowed per switch.

- Regular traffic is disrupted on the ingress UNI port that is used to generate the test traffic. However, traffic on other UNI ports associated with the same SAP profile is not disrupted. Therefore, running the test on a UNI port that is not in use is recommended.
- For bidirectional test the role of the destination switch must be configured as loopback.
- Multicast and broadcast address must not be configured for bidirectional test.
- For the bidirectional test it is mandatory to configure the remote sys MAC address and activate the remote-fetch-stats option while starting the test.

Configuring a CPE Test Profile

This section describes how to configure the following CPE test head example, which includes defining the test profile on the generator and analyzer switch. The configuration steps described in this section also provide a tutorial for how to use the OmniSwitch CLI to configure a CPE test.

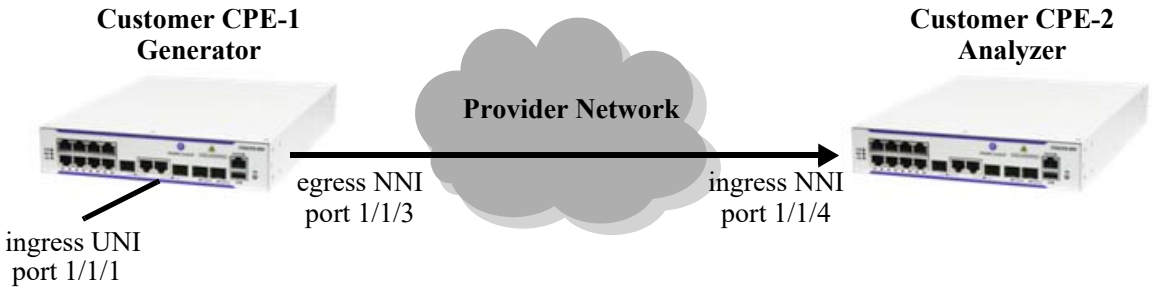


Figure 41-2 : Configuring a CPE Test Profile

To configure the test setup in the above example:

1 Configure the test profile name and an optional description on the generator (CPE-1 switch) and analyzer (CPE-2 switch) using the **test-oam** command. For example:

```
-> test-oam 100M_L2 descr "60 sec 100MB L2 test"
```

When the "100M_L2" test is created, a profile associated with this name is automatically created. This initial profile contains default parameter settings, where applicable. However, in some cases the default values are set to zero as a placeholder, but these parameters require additional configuration.

2 Configure the source (generator) and destination (analyzer) endpoints on CPE-1 and CPE-2 using the **test-oam direction** command. For example:

```
-> test-oam 100M_L2 src-endpoint "CPE-1" dst-endpoint "CPE-2"
```

The endpoint is identified using the DNS host name for the switch. In this example, "CPE-1" and "CPE-2" are the configured host names for the generator and analyze switch.

3 Configure the service VLAN and the source and destination MAC for the test frame on CPE-1 and CPE-2 using the **test-oam vlan test-frame** command. For example:

```
-> test-oam 100M_L2 vlan 100 test-frame src-mac 00:00:00:11:11:11 dst-mac
00:00:00:22:22:22
```

4 Configure CPE-1 as the generator switch using the **test-oam role** command. For example:

```
-> test-oam 100M_L2 role generator
```

Use this command with the **generator** option on the CPE-1 switch. This will configure the role parameter in the "100M_L2" test profile that resides on CPE-1.

5 Configure CPE-2 as the analyzer switch using the **test-oam role** command. For example:

```
-> test-oam 100M_L2 role analyzer
```

Use this command with the **analyzer** option on the CPE-2 switch. This will configure the role parameter in the "100M_L2" test profile that resides on CPE-2.

Note that a switch can only serve as the generator or the analyzer for any given test.

6 Configure port 1/1/1 on CPE-1 as the port on which the test is run, using the **test-oam port** command. For example:

```
-> test-oam 100M_L2 port 1/1/1
```

This is the ingress UNI port that will generate test packets. The packets are then subject to the SAP profile and QoS policies that are associated with the port.

7 Configure the test duration, rate, and size of the test packet on CPE-1 using the **test-oam duration rate packet-size** command. For example:

```
-> test-oam 100M_L2 duration 100 rate 100m packet-size 1518
```

The test duration is the length of time, in seconds, that the test will run. The rate determines the rate at which packets are generated, in bps or Mbps. The packet size specifies the size of the test packet that is generated.

8 Configure a Layer 2 or Layer 3 test frame on CPE-1 using the **test-oam frame** command. The type of test needed determines the type of frame that is configured for the test. If a Layer 2 test is required, configure a Layer 2 frame type; if a Layer 3 test is required, configure a Layer 2 frame type. For example:

To configure a Layer 2 test frame, specify a hexadecimal value for the Ether type.

```
-> test-oam 100M_L2 frame vlan-tag 20 priority 5 ether-type 0x8101 data-pattern
0xabcd
```

To configure a Layer 3 test frame, specify the **ipv4** keyword for the Ether type.

```
-> test-oam 100M_IP frame vlan-tag 10 priority 5 ether-type ipv4 src-ip
10.10.10.111 dst-ip 10.10.10.222
```

See the **test-oam frame** command page in the *OmniSwitch AOS Release 8 CLI Reference Guide* for frame type parameter requirements and definitions.

The following provides a summary of the CLI commands used in the configuration example:

CPE-1 Generator	CPE-2 Analyzer
test-oam 100M_L2 descr "60 sec 100MB L2 Test"	**test-oam 100M_L2 descr "60 sec 100MB L2 Test"**
test-oam 100M_L2 src-endpoint CPE-1 dst-endpoint CPE-2	**test-oam 100M_L2 src-endpoint CPE-1 dst-endpoint CPE-2**
test-oam 100M_L2 vlan 100 test-frame src-mac 00:00:00:11:11:11 dst-mac 00:00:00:22:22:22	**test-oam 100M_L2 vlan 100 test-frame src-mac 00:00:00:11:11:11 dst-mac 00:00:00:22:22:22**

CPE-1 Generator	CPE-2 Analyzer
test-oam 100M_L2 role generator	**test-oam 100M_L2 role analyzer**
test-oam 100M_L2 port 1/1/4	
test-oam 100M_L2 duration 100 rate 100m packet-size 1518	
test-oam 100M_L2 frame vlan-tag 20 priority 5 ether-type 0x8101 data-pattern 0xabcd	

Refer to the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about these commands.

Configuring the L2 SAA Test

The L2-SAA test allows to measure the Round Trip Time (RTT) and Jitter during the test head operation. The L2-SAA test is performed between two OmniSwitch. The test can be run in parallel with the other CPE tests.

The L2-SAA test can also be configured for continuous monitoring of the network performance between the devices. The network performance is monitored by continuous injections of L2-SAA packets throughout the tests which generates and analyze network performance.

To configure the L2-SAA test, use the **test-oam l2-saa** command. For example:

```
-> test-oam test1 l2-saa priority 5 count 5 interval 1000 size 100 drop-eligi-
ble false
```

Note. The CPE test-oam string must be configured before using it in the L2-SAA test. The L2-SAA test derives the source MAC address, destination MAC address, and the VLAN ID from the test-oam configuration of the individual test frames.

To run the L2-SAA test continuously until the test-oam session ends, use the **continuous** parameter. For example:

```
-> test-oam test1 l2-saa continuous priority 5 interval 1000 size 100 drop-
eligible false
```

On receiving the SAA reply for every frame, the minimum RTT, maximum RTT, total RTT, minimum Jitter, maximum Jitter, total Jitter and number of packets received will be calculated and stored in a global buffer to analyze the network performance between the devices.

Use the **show test-oam** command to view the L2-SAA configuration details.

Running a CPE Test

A CPE test is started first on the analyzer switch and then on the generator switch using the **start** form of the **test-oam l2-saa** command. For example:

```
-> test-oam 100M_L2 start
```

This command also includes the following optional parameters used to specify runtime (active) values for the specified test:

- **vlan**—the service VLAN to use for the test.
- **port**—the port on which the test will generate test frames.
- **packet-size**—the size of the test frame to transmit.
- **fetch-remote-stats**—Triggers the test at the remote device from the generator. The statistics are collected during the test and the test is stopped after receiving the test results. The fetch-remote-stats parameter must be used while starting a bidirectional test.

When one or more of these runtime parameters are specified with the **test-oam start** command, the parameter value is used instead of the value configured for the same parameter in the CPE test profile. For example, if the "100M_L2" profile specifies port 1/1/10 for the test, the following command will run the "100M_L2" test on port 1/1/4:

```
-> test-oam 100M_L2 port 1/1/4 start
```

In case the test is a bidirectional test, the **fetch-remote-stats** parameter must be used. For example:

```
-> test-oam "test2" port 1/1/2 start fetch-remote-stats
```

Note. The runtime values specified for any of the optional **test-oam start** command parameters do not overwrite the configured values for the test profile. In addition, if there are no configured values for these parameters in the profile and a runtime value is not specified with the command, the test will not run.

Stopping the CPE Test

An active CPE test is stopped when one of the following two actions occur:

- The duration time configured for the test profile is reached.
- The operator uses the **stop** form of the **test-oam l2-saa** command. For example:

```
-> test-oam 100M_L2 stop
```

Stopping the CPE test on both the generator and analyzer is recommended. The analyzer switch may continue to send out packets attempting to learn the test source if the test is not stopped on the analyzer switch as well.

Verifying the CPE Test Configuration and Results

To display the CPE test configuration and statistics information, use the **show** commands listed below:

test-oam statistics flash-logging	Displays the test configuration and status.
show test-oam statistics	Displays test statistics.
show test-oam saa statistics	Displays the SAA test statistics for all CPE tests or for a specific test name.

The **show test-oam** command displays a summary of CPE test information or more detailed information for a specific test. For example:

```
-> show test-oam tests
Legend: Port: * = Inactive port

Total Test-Ids: 1
Test-Id Port Src-Mac           Dst-Mac          Vlan     Direction     Status      Remote-Sys-Mac
-------+----+-----------------+----------------+-----+---------------+-----------+----------------
test1   1/1/5 00:11:22:12:44:55  00:22:33:12:44:55 1001   bidirectional   running

-> show test-oam Test2
Legend: dei-drop eligible indicator
TEST Parameters for Test2:
  Source Endpoint      : SW1,
  Destination Endpoint : SW2,
  Test Description     : IPV6 Test,
  Direction            : unidirectional,
  Source MAC           : 00:11:22:33:44:55,
  Destination MAC      : 00:22:33:44:55:66,
  Remote Sys MAC       : E8:E7:32:72:01:A4,
  Duration             : 10(secs),
  Vlan                 : 100,
  Role                 : generator,
  Port                 : 1/1/1,
  Tx Rate              : 8k,
  Frame Size           : 100,
  State                : start,
  Status               : running

    Frame Configuration :
      Frame Type       : ipv6,
      Vlan             : 200,
      Priority         : 7,
      Pattern          : 0x0001,
      Dei              : true,
      Source Ip        : 00:00:00:00:10.20.30.50,
      Destination Ip   : 00:00:00:00:10.30.40.60,
      Source Port      : 10,
      Destination Port : 20,
      Next Header      : tcp,
      Hop-Count        : 50,
      Traffic-Class    : 0xff
      Flow-Label       : 0x0

    L2-SAA Configuration :
      L2-SAA Count               : 7
      L2-SAA Interval            : 1000
      L2-SAA DE                  : TRUE
```

```
L2-SAA Payload Size       : 66
L2-SAA Priority           : 0
```

The **show test-oam statistics** command displays packet counts for the number of test packets transmitted and received. For example:

```
-> show test-oam statistics
Test-Id          TX-Ingress   TX-Egress    RX-Ingress Remote-Stats Throughput(Mbps)
--------------+------------+------------+------------+------------+-----------
Test1              1200366      1200366          0        1200366      8
Test2                0            0          1200366     1200366      8
Test3              95553        95553         95553       95553       7.33
```

The packet counts displayed are based on the role the switch plays for the specific test. For example, “Test1” shows statistics for **TX-Ingress** (packets transmitted on ingress UNI) and **TX-Egress** (packets transmitted on egress NNI), but not for **RX-Ingress** (packets received on ingress NNI). This is because the **show** command was performed on the generator switch for “Test1”. The “Test2” display output only for shows statistics for **RX-Ingress** because the switch is the analyzer for “Test2”. The “Test3”displays the statistics for r**emote** test, the number of test frames received by the analyzer/loopback and fetched by the generator device. TX-Ingress, TX-Egress, RX-Ingress, Remote-Stats, and Throughput (Mbps). **Throughput (Mbps)**, displays the traffic throughput of the test.

To verify the received test packet count for “Test1”, use the **show test-oam statistics** command on the analyzer switch. To verify the transmitted test packet count for “Test2”, use the same **show** command on the generator switch.

Note. For more information about the resulting display from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring CPE Test Group

The Customer Provider Edge (CPE) Test Head traffic generator and analyzer is a Test-OAM (Operation, Administration, and Maintenance) tool used in the Metro Ethernet Network to validate the customer Service Level Agreements (SLA). This functionality allows the operator to validate the Metro Ethernet Network between customer end points, which is critical when provisioning or troubleshooting network services.

The following information describes the CPE Test Group multi-test feature and how to configure it through the Command Line Interface (CLI):

- "Quick Steps for Configuring CPE Test Group" on page 41-14
- "CPE Test Group Overview" on page 41-17
- "CPE Test Group Configuration Overview" on page 41-18
- "Configuring a CPE Test Group Profile" on page 41-20
- "Running a CPE Test Group test" on page 41-22
- "Verifying the CPE Test Group Configuration and Results" on page 41-23
- "CPE Test Head Advanced Configuration" on page 41-25
- "Sample Test Configurations" on page 41-27

Quick Steps for Configuring CPE Test Group

The following steps provide a quick tutorial on how to configure a CPE test group and run the CPE test group. Each step describes a specific operation and provides the CLI command syntax that is used to perform that operation.

Configure the CPE Test Group Profile

The CPE test group profile is configured on both the generator and analyzer switch. Steps 2 through 6 configures profile parameters common to both the generator and analyzer switch. Steps 7 through 9 configures profile parameters required only for the generator.

1 Configure the feeder port globally in the system to feed the test traffic to generator port, use the **test-oam feeder** command. For example:

```
-> test-oam feeder-port 1/1/4
```

2 Configure the name for the CPE test group, use the **test-oam group** command. For example:

```
-> test-oam group Testgroup1 descr First-testgroup
```

3 Configure the list of tests that need to be added in the CPE test group, use the **test-oam group tests** command. For example:

```
-> test-oam group Testgroup1 tests test1 test2 test3 test4 test5 test6 test7
test8
```

4 Configure the source and destination end point for the CPE test group, use the **test-oam group src-endpoint dst-endpoint** command. For example:

```
-> test-oam group Testgroup1 src-endpoint SW1

-> test-oam group Testgroup1 dst-endpoint SW2
```

5 Configure the test direction using the **test-oam group direction** command. For example:

```
-> test-oam group Testgroup1 direction bidirectional
```

6 Configure the required role for the switch using the **test-oam group role** command. For example:

```
-> test-oam group Testgroup1 role generator
```

Note. For bidirectional test the role of the destination switch must be configured as loopback.
Direction cannot be set Bidirectional when role is Analyzer and vice-versa.

---

7 Configure the CPE test group port on the generator switch using the **test-oam group port** command. For example:

```
-> test-oam group Testgroup1 port 1/1/2
```

8 Configure the CPE test group duration and rate using the **test-oam group duration rate** command. For example:

```
-> test-oam group Testgroup1 duration 10 rate 8 kbps
```

9 Configure the remote sys mac for the CPE group using the **test-oam group remote-sys-mac** command. This configuration is mandatory in case of bidirectional test. For example:

```
-> test-oam group "Testgroup1" remote-sys-mac E8:E7:32:32:A6:EE
```

Running the CPE Test Group test

1 Start the test on the analyzer switch first and then on the generator switch using the **test-oam group start stop** command. For example:

```
-> test-oam group Testgroup1 port 1/1/2 start

-> test-oam group Testgroup1 start
```

When the test runs for the amount of time specified in the test duration, the test automatically stops.

In case the test is a bidirectional test, the **fetch-remote-stats** parameter must be used. For example:

```
-> test-oam group Testgroup1 port 1/1/2 start fetch-remote-stats
```

2 To stop an active test from running, use the **test-oam group remote-sys-mac** command. For example:

```
-> test-oam group Testgroup1 stop
```

Note. Verify the CPE test group configuration and status with the **show test-oam group** command. For example:

```
-> show test-oam group tests

Total Test-Groups: 2
Feeder Port       : none
Test-Group  Port  Duration     Rate    Nb of   Direction      Status        Remote-Sys-Mac
                   (secs)              Flows
----------+-----+----------+---------+-----+--------------+------------+--------------------
Testgroup1 none     5          -         2   unidirectional  not-started  00:00:00:00:00:00
Testgroup2 none     5          -         3   unidirectional  not-started  00:00:00:00:00:00


   -> show test-oam group Testgroup1
Legend: Port: * = Inactive port

TEST Parameters for Testgroup1:
  Source Endpoint          : SW1,
  Destination Endpoint     : SW2,
  Test Group Description   : first-testgroup,
  Direction                : bidirectional,
  Role                     : generator,
  Tx Rate                  : 10m,
  Duration                 : 60 (secs),
  Port                     : 1/1/1,
  State                    : stop,
  Status                   : ended,
  Remote Sys MAC           : E8:E7:32:32:A6:EE
    Flow 1:
      Test Name            : test1,
      Vlan                 : 1001,
      Tx Rate              : 10m,
      Source MAC           : 00:11:22:12:44:55,
      Destination MAC      : 00:22:33:12:55:66,
```

```
      Remote Sys MAC       : E8:E7:32:32:A6:EE,
      Frame Size           : 100,
      L2-SAA DE            : False,
      L2-SAA Payload Size  : 100,
      L2-SAA Count         : 5,
      L2-SAA Interval      : 1000,
      L2-SAA Priority      : 0
    Flow 2:
      Test Name            : test2,
      Vlan                 : 1001,
      Tx Rate              : 10m,
      Source MAC           : 00:11:22:13:44:55,
      Destination MAC      : 00:22:33:13:55:66,
      Remote Sys MAC       : E8:E7:32:32:A6:EE,
      Frame Size           : 100,
      L2-SAA DE            : False,
      L2-SAA Payload Size  : 100,
      L2-SAA Count         : 5,
      L2-SAA Interval      : 1000,
      L2-SAA Priority      : 0
```

To verify test results, use the **show test-oam group** command. For example:

```
-> show test-oam group statistics
Test-Group Flow    TX-Ingress   TX-Egress    RX-Ingress    Remote-Stats Throughput(Mbps)
----------+------+------------+------------+-------------+-------------+----------------
Testgroup1 test1  309911       309911        309911        309911            4.13
Testgroup1 test2  309730       309730        309730        309730            4.13

-> show test-oam group saa statistics
Test-Group Flow   Time of last run       RTT  RTT    RTT   Jitter Jitter Jitter Packets Description
                   Min        Avg       Max  Min    Avg    Max    Sent   Rcvd
----------+----+---------------------+-----+-----+------+------+------+------+------+------+-------------
Testgroup1 test1 2001-01-10,05:02:59.0 108    111    114      2       3        4       5     first-test-
group
```

To clear test statistics, use the **test-oam group remote-sys-mac** command. For example:

```
-> clear test-oam group Testgroup1 statistics
```

This clears all the statistics related to “Testgroup1”.

```
-> clear test-oam group statistics
```

This will clear the statistics for all the groups configured.

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about these commands.

CPE Test Group Overview

The OmniSwitch CPE Test group feature provides a remote test generator and analyzer capability for testing and validating the Multi-CoS customer Ethernet service domain from end-to-end. The feature supports up to four concurrent test flows. The OmniSwitch CPE Test group feature allows the service provider to perform the following tasks without the need for an external test head device:

- Generate specific flow-based traffic across the customer's Ethernet Virtual Circuit (EVC) to help identify flow-based issues.
- Identify the impact of QoS settings (SAP profile or QoS policies) on the overall traffic.
- Confirm throughput across the provider network.
- Debug flow-specific traffic forwarding across the provider network.
- Analyze the behavior of various user-defined traffic patterns across the provider network.
- Perform the handover testing after initial deployment.
- Perform on-demand testing and results monitoring using a central entity.

The OmniSwitch implementation of CPE Test group supports the ability to run unidirectional and bidirectional, ingress tests. Test setup involves configuring one CPE switch as the generator and a remote switch as the analyzer/loopback.

The following diagram shows an example of an OmniSwitch CPE Test Group configuration:

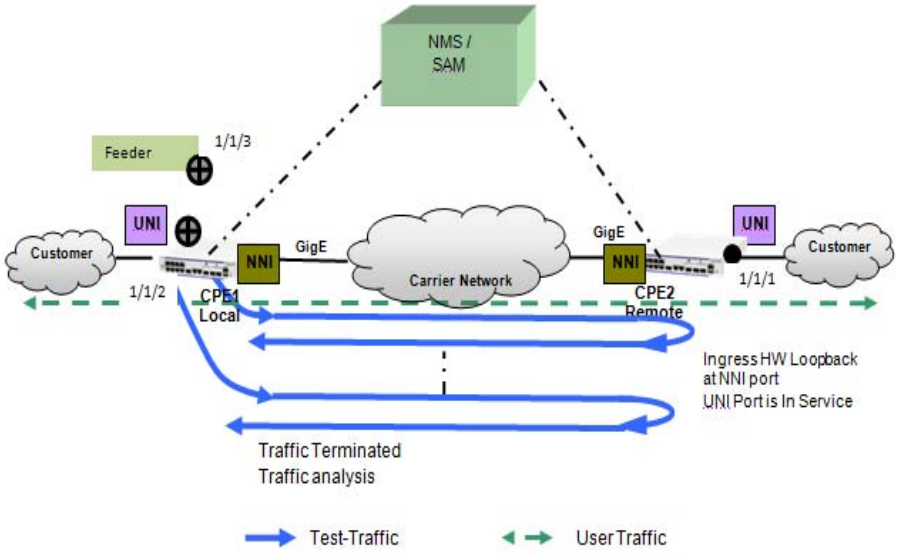


Figure 41-3 : CPE Test group Example - Unidirectional, Ingress Test

In this example:

1 A feeder port must be configured in the system to feed the traffic to the generator. The feeder port is required while running a CPE test group.

2 The CPE test group is started first on the analyzer switch and then on the generator switch. The analyzer switch sends packets to the generator switch to learn the source.

3 A configurable amount of traffic is generated and counted on the ingress UNI port of the generator switch, as if the traffic was generated from a test head connected to the UNI port. This subjects the test traffic to the ingress UNI SAP profile policies.

4 Traffic is counted and sent out on the SAP NNI port. This subjects the test frames to the egress NNI QoS policies.

5 Test frames are forwarded through the provider network over the customer EVC to the ingress NNI on the analyzer switch, where the packets are received and counted. Note that test frames are dropped after they are counted.

6 CPE Test group CLI **show** commands are used on the generator and analyzer switches to display and verify CPE test group statistics, such as packets transmitted and received.

Note. The CPE test is non-disruptive to traffic running on other UNI ports that are associated with the same SAP profile as the test UNI port. All UNI and NNI ports, including CPE test ports, are subject to any SAP profile or QoS configuration associated with the SAP profile. This is important to consider when analyzing test results.

CPE Test Group Configuration Overview

CPE Test Group configuration is done using a test profile to define test attributes. Configuring a test profile is required on both the generator and analyzer switch. Not all test profile information is required for both switches. For example, the profile on the generator switch must contain a port number to identify the UNI port on which the test will run, but a port number is not required for the analyzer profile.

The following table provides a list of CPE test group parameters and identifies if the parameter is required on the generator, analyzer, or both. Also included is the CLI command used to configure the parameter.

CPE Test group Parameters	Generator Switch	Analyzer/ Loopback Switch	CLI Command
Profile name	Yes	Yes	**test-oam group**
Source and destination endpoints	Yes	Yes	**test-oam group src-endpoint dst-endpoint**
Test-oam role (generator or analyzer or loopback)	Yes	Yes	**test-oam group role**
UNI port for test packet generation	Yes	No	**test-oam group port**
Test-oam duration and rate	Yes	No	**test-oam group duration rate**
Remote Sys MAC	Yes	No	**test-oam group remote-sys-mac**

Configuration Guidelines

Consider the following guidelines when configuring the OmniSwitch CPE Test group:

- Make sure the same CPE test group name (test ID) is used on the generator and analyzer switch.
- A switch can only perform one role (generator or analyzer) for a specific test.
- Each test which will be configured in the list of tests in the CPE test group that needs to run concurrently must be configured before adding in the list.
- Each flow is properly configured to be classified into the correct CoS or QoS profile.
- The sum of bandwidth of the grouped test streams must not exceed the supported line-rate of 100 Mbps for copper port and 1 Gig for fiber port.
- Only one CPE test group can be active for the switch at any given time.
- Up to 32 CPE test groups are allowed per switch.
- The feeder port must be configured to start a CPE test group.
- The VLAN used for a CPE test group must be a service VLAN.
- Each test in a CPE test group must have a unique VLAN, source mac-address, and destination mac-address.
- The modification to the test which is part of the active CPE test group is not allowed.
- The CPE test group supports four test flows that can run concurrently.
- For bidirectional test, the role of the destination switch must be configured as loopback.
- Multicast and broadcast address must not be configured for bidirectional test.

Configuring a CPE Test Group Profile

This section describes how to configure the following CPE test group example, which includes defining the CPE test group profile on the generator and analyzer switch. The configuration steps described in this section also provide a tutorial for how to use the OmniSwitch CLI to configure a CPE test group.

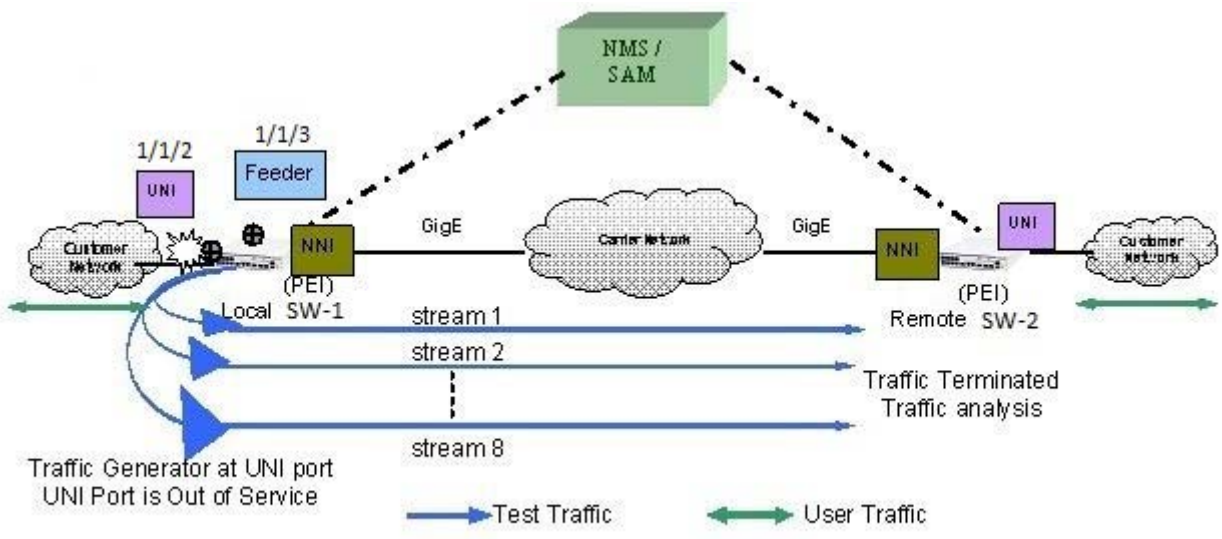


Figure 41-4 : Configuring a CPE Test Group Profile

To configure the test setup shown in the above figure:

1 Configure the feeder port globally in the system to feed the test traffic to generator port, use the **test-oam feeder** command. For example:

```
-> test-oam feeder-port 1/1/3
```

The configured feeder port 1/1/3 will feed the test traffic from the CPE test group to the generator port.

2 Configure the CPE test group profile name and an optional description on the generator (SW-1 switch) and analyzer (SW-2 switch) using the **test-oam group** command. For example:

```
-> test-oam group Testgroup1 descr first-testgroup
```

When the “Testgroup1” CPE test group is created, a profile associated with this name is automatically created.

3 Configure the list of CPE test group tests that need to be added in the CPE test group using the **test-oam group tests** command. For example:

```
-> test-oam group Testgroup1 tests test1 test2 test3 test4 test5 test6 test7
test8
```

The configured list of CPE test group tests will run concurrently when the CPE test group Testgroup1 is started.

4 Configure the source (generator) and destination (analyzer) endpoints on SW-1 and SW-2 using the **test-oam group src-endpoint dst-endpoint** command. For example:

```
-> test-oam group Testgroup1 src-endpoint SW1 dst-endpoint SW2
```

The endpoint is identified using the DNS host name for the switch. In this example, “SW-1” and “SW-2” are the configured host names for the generator and analyze switch.

5 Configure SW-1 as the generator switch using the **test-oam group role** command. For example:

```
-> test-oam group Testgroup1 role generator
```

Use this command with the **generator** option on the SW-1 switch. This will configure the role parameter in the “Testgroup1” CPE test group profile that resides on SW-1.

6 Configure SW-2 as the analyzer switch using the **test-oam group role** command. For example:

```
-> test-oam group Testgroup1 role analyzer
```

Use this command with the **analyzer** option on the SW-2 switch. This will configure the role parameter in the “Testgroup1” CPE test group profile that resides on SW-2.

Note that a switch can only serve as the generator or the analyzer for any given test.

7 Configures the port in SW-1 on which the CPE test group test will run, using the **test-oam group port** command. For example:

```
-> test-oam group Testgroup1 port 1/1/2
```

This is the ingress UNI port that will generate test packets. The packets are then subject to the SAP profile and QoS policies that are associated with the port.

8 Configure the test duration and rate of the CPE test group packet on SW-1 using the **test-oam group duration rate** command. For example:

```
-> test-oam group Testgroup1 duration 20 rate 8m
```

The test duration is the length of time, in seconds, that the test will run. The rate determines the rate at which packets are generated, in kbps or mbps. The group rate configuration is optional. The test bandwidth is considered by default if the group rate is not configured.

The following table provides a summary of the CLI commands used in the configuration example:

SW-1 Generator	SW-2 Analyzer
test-oam group Testgroup1 descr first-testgroup	**test-oam group Testgroup1 descr first-testgroup**
test-oam group Testgroup1 tests test1 test2 test3 test4 test5 test6 test7 test8	**test-oam group Testgroup1 tests test1 test2 test3 test4 test5 test6 test7 test8**
test-oam group Testgroup1 src-endpoint SW1 dst-endpoint SW2	**test-oam group Testgroup1 src-endpoint SW1 dst-endpoint SW2**
test-oam group Testgroup1 role generator	**test-oam group Testgroup1 role analyzer**
test-oam group Testgroup1 duration 20	**test-oam group Testgroup1 duration 20**
test-oam group Testgroup1 port 1/1/2	
test-oam group Testgroup1 rate 8m	

Refer to the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about these commands.

Running a CPE Test Group test

A CPE test is started first on the analyzer switch and then on the generator switch using the **test-oam group start stop** command. For example:

```
-> test-oam group Testgroup1 start
```

This command also includes the following optional parameter used to specify runtime (active) values for the specified test:

port—the port on which the test will generate test frames.

When this runtime parameter is specified with the **test-oam group start stop** command, the parameter value is used instead of the value configured for the same parameter in the CPE test group profile. For example, if the "Testgroup1" profile specifies port 1/1/10 for the test, the following command will run the "Testgroup1" test on port 1/1/4:

```
-> test-oam group Testgroup1 port 1/1/4 start
```

Note. The runtime values specified for any of the optional **test-oam group start** command parameters do not overwrite the configured values for the test profile. In addition, if there are no configured values for these parameters in the profile and a runtime value is not specified with the command, the test will not run.

Stopping the CPE Test Group test

An active CPE test group test is stopped when one of the following two actions occur:

- The duration time configured for the test profile is reached.
- The operator uses the **test-oam group remote-sys-mac** command. For example:

```
-> test-oam group Testgroup1 stop
```

Stopping the CPE test group on both the generator and analyzer is recommended. The analyzer switch will continue to send out packets attempting to learn the test source if the test is not stopped on the analyzer switch as well.

Verifying the CPE Test Group Configuration and Results

To display the CPE test group configuration and statistics information, use the **show** commands listed below:

show test-oam group	Displays the configuration and status of the CPE test groups.
show test-oam group	Displays the statistics for all CPE test groups or for a specific CPE test group.
show test-oam group	Displays the SAA test statistics for all CPE test groups or for a specific test name.

The **show test-oam group** command displays the configuration and status of the CPE test groups. For example:

```
-> show test-oam group tests

Total Test-Groups: 2
Feeder Port      : none
Test-Group  Port  Duration     Rate    Nb of   Direction      Status        Remote-Sys-Mac
                   (secs)              Flows
----------+-----+----------+---------+-----+---------------+------------+--------------------
Testgroup1 none    5          -         2   unidirectional  not-started  00:00:00:00:00:00
Testgroup2 none    5          -         3   unidirectional  not-started  00:00:00:00:00:00


-> show test-oam group TestGroup2

TEST Parameters for TestGroup2:
  Source Endpoint: SW1,
  Destination Endpoint: SW2,
  Test Group Description: DEFAULT,
  Direction: unidirectional,
  Role: generator,
  Tx Rate : -,
  Duration : 20 (secs),
  Port: 1/1/2,
  State: stop,
  Status: stopped

Flow1:
      Test Name : test_1,
      Vlan: 1001
      Tx Rate    : 1M,
      Source MAC: 00:00:00:00:01:01,
      Destination MAC: 00:00:00:00:01:02,
      Remote Sys MAC : E8:E7:32:72:01:A4,
      Frame size: 64,
      L2-SAA DE            : False,
      L2-SAA Payload Size : 100,
      L2-SAA Count         : 5,
      L2-SAA Interval      : 1000,
      L2-SAA Priority      : 0

Flow2:
      Test Name : test_2,
      Vlan: 1002
      Tx Rate    : 10M,
```

```
      Source MAC: 00:00:00:00:02:01,
      Destination MAC: 00:00:00:00:02:02,
      Remote Sys MAC : E8:E7:32:72:01:A4,
      Frame size: 1518,
      L2-SAA DE           : False,
      L2-SAA Payload Size : 100,
      L2-SAA Count        : 5,
      L2-SAA Interval     : 1000,
      L2-SAA Priority     : 0

Flow3:
      Test Name : test_3,
      Vlan: 1003
      Tx Rate: 15M,
      Source MAC: 00:00:00:00:03:01,
      Destination MAC: 00:00:00:00:03:02,
      Remote Sys MAC : E8:E7:32:72:01:A4,
      Frame size: 1518,
      L2-SAA DE           : False,
      L2-SAA Payload Size : 100,
      L2-SAA Count        : 5,
      L2-SAA Interval     : 1000,
      L2-SAA Priority     : 0

Flow4:
      Test Name : test_4,
      Vlan: 1004
      Tx Rate: 5M,
      Source MAC: 00:00:00:00:04:01,
      Destination MAC: 00:00:00:00:04:02,
      Remote Sys MAC : E8:E7:32:72:01:A4,
      Frame size: 1518,
      L2-SAA DE           : False,
      L2-SAA Payload Size : 100,
      L2-SAA Count        : 5,
      L2-SAA Interval     : 1000,
      L2-SAA Priority     : 0
```

The **show test-oam group** command displays the statistics for all CPE test groups or for a specific CPE test group. For example:

```
-> show test-oam group statistics

Test-Group  Flow    TX-Ingress   TX-Egress    RX-Ingress  Remote-Stats   Throughput(Mbs)
-----------+------+------------+------------+-----------+-------------+----------------------
TestGroup1 flow1    19017        19017          0         19017            9.98
TestGroup1 flow2    19017        19017          0         19017            9.98
TestGroup1 flow3    19017        19017          0         19017            9.98
TestGroup1 flow4    19017        19017          0         19017            9.98
TestGroup1 flow5    19017        19017          0         19017            9.98
TestGroup1 flow6    19017        19017          0         19017            9.98
TestGroup1 flow7    19017        19017          0         19017            9.98
TestGroup1 flow8    19017        19017          0         19017            9.98
TestGroup2 flow1    19017        19017          0         0
TestGroup2 flow2    19017        19017          0         0
TestGroup2 flow3    19017        19017          0         0
TestGroup2 flow4    19017        19017          0         0
TestGroup3 flow1    19017        19017          0         0
TestGroup4 flow8    19017        19017          0         19017            9.98
```

The packet counts displayed are based on the role the switch plays for the specific test. For example, “TestGroup1” shows statistics for **TX-Ingress** (packets transmitted on ingress UNI) and **TX-Egress** (packets transmitted on egress NNI), but not for **RX-Ingress** (packets received on ingress NNI). This is because the **show** command was performed on the generator switch for “TestGroup1”.

To verify the received test packet count for “TestGroup1”, use the **show test-oam group** command on the analyzer switch.

Note. For more information about the resulting display from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

CPE Test Head Advanced Configuration

Running L2 SAA test

The CPE test can be used to measure the Round Trip Time (RTT) and Jitter by using the **test-oam l2-saa** command. The L2 SAA test will run along with the data traffic test. The test results are captured at the generator switch.

For example:

```
-> test-oam test1 l2-saa count 8 size 120 priority 6 interval 900 drop-eligible
true
```

Note. Use the **show test-oam saa statistics** command to view the test results.

Configuring Remote Sys MAC

The CPE test allows configuring a remote device to receive the test OAM messages on the generator side. The generator device can gather the test OAM messages from the remote device and store it in the local data base.

Configuring the Remote Sys MAC is mandatory for bidirectional test and optional for unidirectional test.

In case of single stream test, use the **test-oam remote-sys-mac** command to configure the remote device to receive the test OAM messages. For example:

```
-> test-oam Test1 remote-sys-mac 00:e0:b1:7c:7a:fa
```

In case of multi stream test, use the **test-oam group remote-sys-mac** command to configure the remote device to receive the test OAM messages. For example:

```
-> test-oam group Testgroup1 remote-sys-mac 00:e0:b1:7c:7a:fa
```

Saving the test results on the /flash

The test results can be stored on the /flash directory of the switch. The test information is appended at the end of the default text file. Two files are used to maintain the test statistics on the /flash directory active file (testoamActiveStats.txt) and inactive file (testoamInactiveStats.txt).

The current test statistics will be stored in the active file. When there is no space in the active file to store the test statistics, the active file is made inactive and the inactive file is made active and the stats are written by overwriting the old data.

Use the **test-oam statistics flash-logging** command to enable storing of the test information on the /flash. For example:

```
-> test-oam statistics flash-logging enable
```

Note. Use the **more** command to read the test results stored on the switch.

Sample Test Configurations

Sample Unidirectional Test Configuration

The following scenario represents a sample unidirectional test configuration:

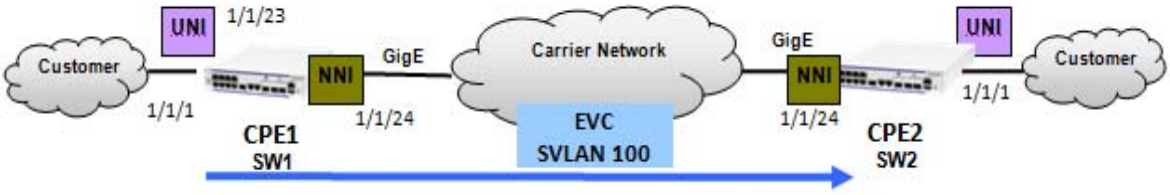


Figure 41-5 : Sample Unidirectional Test Configuration

The CLI configuration for the displayed scenario is shown in the following table:

CPE1 SW-1 Generator	CPE2 SW-2 Analyzer
test-oam Test1	**test-oam Test1**
test-oam Test1 src-endpoint SW1	**test-oam Test1 src-endpoint SW1**
test-oam Test1 dst-endpoint SW2	**test-oam Test1 dst-endpoint SW2**
test-oam Test1 direction unidirectional	**test-oam Test1 direction unidirectional**
test-oam Test1 test-frame src-mac 00:00:00:00:00:01	**test-oam Test1 test-frame src-mac 00:00:00:00:00:01**
test-oam Test1 test-frame dst-mac 00:00:00:00:00:02	**test-oam Test1 test-frame dst-mac 00:00:00:00:00:02**
test-oam Test1 vlan 100	**test-oam Test1 vlan 100**
test-oam Test1 role generator	**test-oam Test1 role analyzer**
test-oam Test1 port 1/1/1	
test-oam Test1 frame vlan-tag 10 priority 5 ether-type ipv4 src-ip 1.1.1.1 dst-ip 2.2.2.2 src-port 2000 dst-port 4000	**test-oam Test1 frame vlan-tag 10 priority 5 ether-type ipv4 src-ip 1.1.1.1 dst-ip 2.2.2.2 src-port 2000 dst-port 4000**
test-oam Test1 packet-size 64	
test-oam Test1 rate 10Mbps	
test-oam Test1 duration 10	

Sample Bidirectional Test Configuration

The following scenario represents a sample bidirectional test configuration:

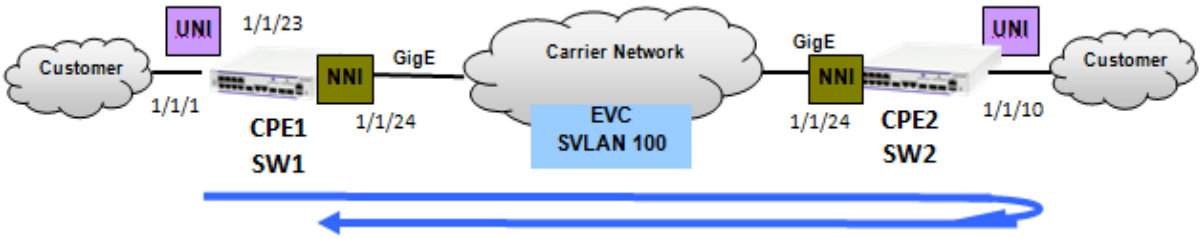


Figure 41-6 : Sample Bidirectional Test Configuration

The CLI configuration for the displayed scenario is shown in the following table:

CPE1 SW-1 Generator	CPE2 SW-2 Loopback
test-oam Test2	**test-oam Test2**
test-oam Test2 src-endpoint SW1	**test-oam Test2 src-endpoint SW1**
test-oam Test2 dst-endpoint SW2	**test-oam Test2 dst-endpoint SW2**
test-oam Test2 direction bidirectional	**test-oam Test2 direction bidirectional**
test-oam Test2 test-frame src-mac 00:00:00:00:00:01	**test-oam Test2 test-frame src-mac 00:00:00:00:00:01**
test-oam Test2 test-frame dst-mac 00:00:00:00:00:02	**test-oam Test2 test-frame dst-mac 00:00:00:00:00:02**
test-oam Test2 vlan 100	**test-oam Test2 vlan 100**
test-oam Test2 role generator	**test-oam Test2 role loopback**
test-oam Test2 remote-sys-mac E8:E7:32:32:A6:EE	**test-oam Test2 port 1/1/10**
test-oam Test2 port 1/1/1	**test-oam Test2 frame vlan-tag 10 priority 5 ether-type ipv4 src-ip 1.1.1.1 dst-ip 2.2.2.2 src-port 2000 dst-port 4000**
test-oam Test2 frame vlan-tag 10 priority 5 ether-type ipv4 src-ip 1.1.1.1 dst-ip 2.2.2.2 src-port 2000 dst-port 4000	**test-oam Test2 port 1/1/10**
test-oam Test2 packet-size 64	
test-oam Test2 rate 10Mbps	
test-oam Test2 duration 10	

Note. For bidirectional test, the role of the destination switch must be configured as loopback.

Sample Bidirectional Multi-stream Test Configuration

The following scenario represents a sample bidirectional multi-stream test configuration:

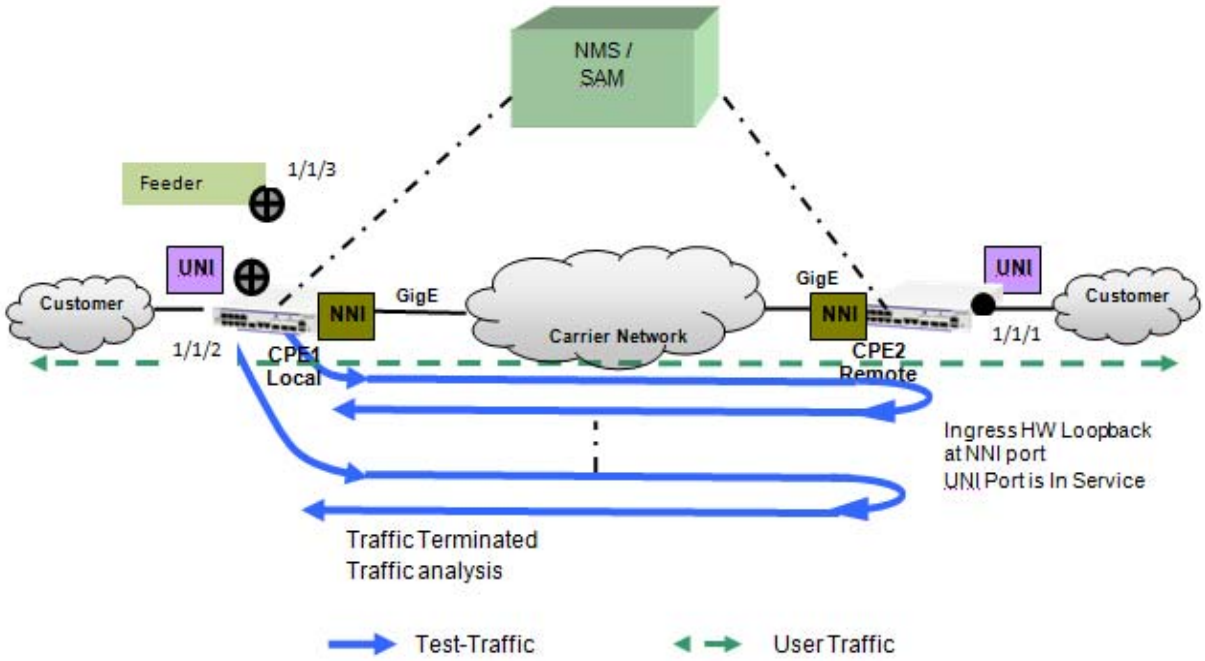


Figure 41-7 : Sample Bidirectional Multi-stream Test Configuration

The CLI configuration for the displayed scenario is shown in the following table:

CPE1 SW-1 Generator	CPE2 SW-2 Loopback
test-oam feeder-port 1/1/3	
test-oam group "testgroup1" descr "first-testgroup"	**test-oam group "testgroup1" descr "first-testgroup"**
test-oam group "testgroup1" tests "test2" "test3"	**test-oam group "testgroup1" tests "test2" "test3"**
test-oam group "testgroup1" direction bidirectional	**test-oam group "testgroup1" direction bidirectional**
test-oam group "testgroup1" src-endpoint SW1 dst-endpoint SW2	**test-oam group "testgroup1" src-endpoint SW1 dst-endpoint SW2**
test-oam group "testgroup1" remote-sys-mac E8:E7:32:32:A6:EE	**test-oam group "testgroup1" remote-sys-mac e8:e7:32:32:a8:9e**
test-oam group "testgroup1" port 1/1/2	**test-oam group "testgroup1" port 1/1/1**
test-oam group "testgroup1" role generator	**test-oam group "testgroup1" role loopback**
test-oam group "testgroup1" duration 10	**test-oam group "testgroup1" duration 10**
test-oam group "testgroup1" tests "test1" "test2" "test3" "test4"	**test-oam group "testgroup1" tests "test1" "test2" "test3" "test4"**

Note. The individual tests must be configured before being added to the test group.
A maximum of four tests can be added in a group.

42 Configuring PPPoE Intermediate Agent

Point-to-Point Protocol over Ethernet (PPPoE) provides the ability to connect a network of hosts over a simple bridging access device to a Remote Access Concentrator (RAC). For example, Broadband Network Gateway. In PPPoE model, each host utilizes its own Point-to-Point Protocol (PPP) stack and the user is presented with a familiar user interface. Access control, billing, and type of service can be configured on a per-user, rather than a per-site, basis.

PPPoE Intermediate Agent (PPPoE-IA) solution is designed for the PPPoE access method and is based on the access node implementing a PPPoE-IA function to insert the access loop identification.

In This Chapter

This chapter describes the PPPoE-IA feature and how to configure it through the Command Line Interface (CLI). CLI commands are used in the configuration examples. For more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

This chapter includes the following:

- "PPPoE-IA Defaults" on page 42-1
- "Quick Steps for Configuring PPPoE-IA" on page 42-2
- "PPPoE Intermediate Agent Overview" on page 42-4
- "Configuring PPPoE-IA" on page 42-5
- "Verifying PPPoE-IA Configuration" on page 42-8

PPPoE-IA Defaults

Following are the PPPoE-IA default values:

Parameter Description	Command	Default Value
PPPoE-IA globally and on ports	**pppoe-ia** **pppoe-ia {port | linkagg}**	Disabled
PPPoE-IA port	**pppoe-ia {trust | client}**	Client
Access-Node-Identifier	**pppoe-ia access-node-id**	Base MAC address of the switch
Circuit-ID	**pppoe-ia circuit-id**	":" (colon) is used as the delimiter
Remote-ID	**pppoe-ia remote-id**	Base MAC address of the switch

Quick Steps for Configuring PPPoE-IA

The following steps provide a quick tutorial on how to configure PPPoE-IA. Each step describes a specific operation and provides the CLI command syntax for performing that operation.

1 Enable PPPoE-IA globally on the switch using the **pppoe-ia** command.

```
-> pppoe-ia enable
```

Note. All PPPoE-IA parameters are configurable irrespective of the global status of PPPoE-IA. It is mandatory to enable PPPoE-IA globally as well as on a port for the PPPoE-IA feature to function.

---

2 Enable PPPoE-IA on a port or a link aggregate port using the **pppoe-ia {port | linkagg}** command. For example, the following command enables PPPoE-IA on port 1/1/1 of the switch.

```
-> pppoe-ia port 1/1/1 enable
```

3 Configure a port or a link aggregate port as trusted or client port for PPPoE-IA using the **pppoe-ia {trust | client}** command. By default, all ports are client ports. For example, the following command configures port 1/1/1 as a trusted port.

```
-> pppoe-ia port 1/1/1 trust
```

Note. The port that is connected to the PPPoE server must be configured as trusted, whereas the port connected to the host must be configured as a client port. Both client and trust ports must be in the same VLAN.

---

4 Configure a format to form an identifier that uniquely identifies an access node globally using the **pppoe-ia access-node-id** command. For example, the following command uses the base MAC address of the switch to identify an access node.

```
-> pppoe-ia access-node-id base-mac
```

5 Configure a Circuit-ID format that forms an identifier that uniquely identifies an access node globally, and an access loop that receives the PADI/PADR/PADT from the user side using the **pppoe-ia circuit-id** command. For example, the following command uses the base MAC address in ASCII format as the Circuit-ID.

```
-> pppoe-ia circuit-id ascii base-mac vlan
```

6 Configure a format to form an identifier that uniquely identifies the user attached to the access loop globally using the **pppoe-ia remote-id** command. For example, the following command uses the user configured string as the format for Remote-ID:

```
-> pppoe-ia remote-id user-string "remote-id-1"
```

Note. To view the global configuration for PPPoE-IA, enter the **show pppoe-ia configuration** command. The PPPoE-IA configuration is displayed as shown:

```
-> show pppoe-ia configuration
Status                      : enabled,
Access Node Identifier
  Access-node-id Format      : system-name,
  Access-node-id String      : vxTarget,
Circuit Identifier
  Circuit-Id Format         : ascii,
  Circuit-id Field1         : system-name,
  Circuit-id Field1 String  : vxTarget,
  Circuit-id Field2         : base-mac,
  Circuit-id Field2 String  : 00:d0:95:ee:fb:02,
  Circuit-id Field3         : interface,
  Circuit-id Field3 String  : ,
  Circuit-id Field4         : none,
  Circuit-id Field4 String  : ,
  Circuit-id Field5         : none,
  Circuit-id Field5 String  : ,
  Circuit-id Delimiter      : "|",
Remote Identifier
  Remote-id Format          : mgnt-address,
  Remote-id String          : 172.21.161.106
```

PPPoE Intermediate Agent Overview

PPPoE Intermediate Agent (PPPoE-IA) solution is designed for the PPPoE access method and is based on the access node implementing a PPPoE intermediate agent function to insert the access loop identification.

Access Node: An access node provides connectivity between the user and the network cloud. Access node aggregates the traffic coming from a user and routes it to the network. In the context of PPPoE-IA, an access node is the switch where the Intermediate Agent (IA) resides.

Access Loop: Access loop signifies the physical connectivity between the Network Interface Device (NID) at the customer premises and the access node. If a user is directly connected to the access node, the access loop can be identified by the interface number (chassis/slot/port). If the user is not directly connected or multiple users are connected to the access node through a single port, access loop for a particular user can be identified as the combination of interface (chassis/slot/port) and customer VLAN (CVLAN).

How PPPoE-IA Works

PPPoE-IA is a means by which the discovery packets of PPPoE are tagged at the access switch of the service provider using Vendor Specific Attributes (VSA) to add the line-specific information at the switch.

The purpose of an IA is to help service provider and the Broadband Network Gateway to distinguish between different end hosts connected over Ethernet to the access switch. The Ethernet frames from different users are appropriately tagged by the IA to provide this distinction. The AOS implementation of PPPoE-IA enables the rate limiting and insertion of VSA tags into the PPPoE Active Discovery (PAD) messages. The tag is allowed to contain information such as the base MAC address of the switch, interface, customer VLAN, system name, and a user-defined string depending on the configuration.

The following example illustrates the network overview for PPPoE IA.

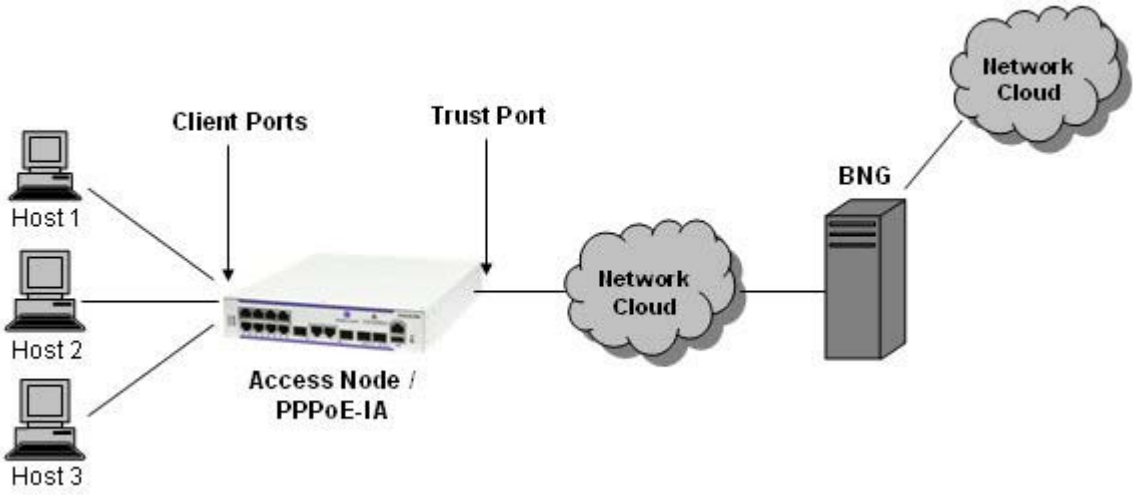


Figure 42-1 : Network overview for PPPoE IA

Configuring PPPoE-IA

This section describes how to configure PPPoE-IA using the CLI commands.

Enabling PPPoE-IA Globally

Enable the PPPoE-IA globally on the switch. By default, PPPoE-IA is disabled globally on the switch.

To enable PPPoE-IA globally on the switch, enter the **pppoe-ia** command at the CLI prompt as shown:

```
-> pppoe-ia enable
```

To disable PPPoE-IA globally on the switch, use disable option as shown:

```
-> pppoe-ia disable
```

Note. All PPPoE-IA parameters are configurable irrespective of the global status of PPPoE-IA. It is mandatory to enable PPPoE-IA globally as well as on a port for the PPPoE-IA to function.

Enabling PPPoE-IA on a Port

Enable or disable PPPoE-IA on a port or a link aggregate port by using **pppoe-ia {port | linkagg}** command. It is mandatory that PPPoE-IA is enabled globally as well as on a port.

For example, to enable PPPoE-IA on port 1/1/1 of the switch, enter:

```
-> pppoe-ia port 1/1/1 enable
```

To disable PPPoE-IA on port 1/1/2, enter:

```
-> pppoe-ia port 1/1/2 disable
```

Note. PPPoE-IA is not supported on port mirroring destination ports, however, the configurations are accepted. PPPoE-IA is not supported on individual ports of an aggregate.

Configuring a Port as Trust or Client

Use **pppoe-ia {trust | client}** command to configure a port or a link aggregate port as trusted or client port. PPPoE-IA must be enabled on a client port as well as a trusted port for the feature to function. By default, all ports are client ports.

The port that is connected to the PPPoE Server must be configured as trusted, whereas the port connected to the host must be configured as a client port.

For example, to configure port 1/1/1 as a trusted port, enter:

```
-> pppoe-ia port 1/1/1 trust
```

For example, to configure link aggregate port 0 as a client port, enter:

```
-> pppoe-ia linkagg 0 client
```

Configuring Access Node Identifier for PPPoE-IA

To configure a format to form an identifier that uniquely identifies an access node, use the **pppoe-ia access-node-id** command.

For example, the following command uses the base MAC address of the switch to identify an access node:

```
-> pppoe-ia access-node-id base-mac
```

For example, the following command uses the user configured string to identify an access node:

```
-> pppoe-ia access-node-id user-string acessnode1
```

If the management address format is used as the Access Node Identifier, then the IP address of the Loopback0 interface (if configured and active) or the first active IP interface address is used as the management address. If none of them are available, IP address '0.0.0.0' is used as management address.

The access-node-identifier can have a maximum of 32 characters. The access-node-identifier longer than 32 characters is truncated to 32 characters when encoded in the VSA tag.

Configuring Circuit Identifier

The **pppoe-ia circuit-id** command globally configures a Circuit-ID format that forms an identifier that uniquely identifies an access node and an access loop on which the PPPoE Active Discovery Initiation (PADI) or PPPoE Active Discovery Request (PADR) or PPPoE Active Discovery Terminate (PADT) is received.

For Circuit-ID, two-format types are supported: default and ascii. The Circuit-ID is formed depending on the format as follows:

Default Circuit ID

default: When the PPPoE Circuit-ID is configured as default, the access-node-id is formed from either of the four supported formats: base-mac, system-name, mgnt-address, or user configurable string.

For example, the following command is used to configure the Circuit-ID as default.

```
-> pppoe-ia circuit-id default
```

When the Circuit-ID is configured as default, the Circuit-ID format in the Circuit-Identifier will display as "ethernet". For more information, see **show pppoe-ia configuration** command in the *OmniSwitch AOS Release 8 CLI Reference Guide*

default ATM: When the PPPoE-IA Circuit-ID format is configured as "default atm" the Circuit-ID encoding happens for "ATM" (Asynchronous Transfer Mode) parameter along with ethernet parameter.

For example, the following command is used to configure the Circuit-ID as "default ATM".

```
-> pppoe-ia circuit-id default atm
```

When the Circuit-ID is configured as default ATM, the Circuit-ID format in the Circuit-Identifier will display as "atm". For more information, see **show pppoe-ia configuration** command in the *OmniSwitch AOS Release 8 CLI Reference Guide*

ASCII Circuit ID

In the ascii Circuit-ID, the fields (maximum of five) are separated by delimiter up to a maximum of 63 characters.

For example, the following command uses the base-mac in ASCII format of the Circuit-ID:

```
-> pppoe-ia circuit-id ascii base-mac vlan
```

Configuring Remote Identifier

The Remote-ID identifies the host attached to the access loop. In AOS implementation, the Remote-ID identifies the access-node (that is, the IA).

The **pppoe-ia remote-id** command globally configures a format to form an identifier that uniquely identifies the user attached to the access loop.

For example, to use the base MAC address as the format for Remote-ID, enter:

```
-> pppoe-ia remote-id base-mac
```

If the management address format is used as the Remote-ID, the IP address of the Loopback0 interface (if configured and active) or the first active IP interface address is used as the management address. If none of them are available, IP address ‘0.0.0.0’ is used as management address.

Verifying PPPoE-IA Configuration

A summary of the commands used for verifying the PPPoE-IA configuration is given here:

show pppoe-ia configuration	Displays the global configuration for PPPoE-IA.
show pppoe-ia {port | linkagg}	Displays the PPPoE-IA configuration for a port, port range or all the ports.
show pppoe-ia statistics	Displays the PPPoE-IA statistics for a port, link aggregate port, port range, or all the ports.

To clear the statistics for all the physical or link-aggregate ports, a single port or a link aggregate port, or a range of physical ports for PPPoE-IA, use the **clear pppoe-ia statistics** command.

For more information about the output details that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

43 Configuring Service Assurance Agent

Service Assurance Agent (SAA) enables customers to assure business-critical applications, as well as services that utilize data, voice, and video. With SAAs, users can verify service guarantees, increase network reliability by validating network performance and proactively identify network issues. SAA uses active monitoring to generate traffic in a continuous, reliable, and predictable manner, thus enabling the measurement of network performance and health.

In This Chapter

This chapter describes the various types of SAAs that can be configured on an OmniSwitch. Configuration procedures described in this chapter include:

- "Configuring an SAA ID" on page 43-5.
- "Configuring a MAC Address Ping SAA" on page 43-6.
- "Configuring an IP Ping SAA" on page 43-6.
- "Configuring an Ethernet OAM SAA" on page 43-6.
- "Starting and Stopping SAAs" on page 43-7
- "Enabling Jitter Calculation in SAAs" on page 43-7
- "Configuring SAA SPB Session Parameters" on page 43-8.
- "Generating an SAA XML History File" on page 43-8.
- "Verifying the SAA Configuration" on page 43-12.

SAA Defaults

There are no SAAs created by default. However, when an agent is configured, the following default parameter values are applied unless otherwise specified:

Parameter Description	Command	Default Value/Comments
Time interval between test iterations	**saa interval**	150 minutes
SAA description	**saa descr**	"DEFAULT"
SAA jitter threshold	**saa jitter-threshold**	0 (disabled)
SAA round-trip-time threshold	**saa rtt-threshold**	0 (disabled)

Quick Steps for Configuring SAA

The following steps provide a quick tutorial on how to configure SAA. Each step describes a specific operation and provides the CLI command syntax for performing that operation.

Creating SAA:

1 Create the base SAAs using the **saa** command. For example:

```
-> saa saa1 description "saa for ip-ping" interval 120 rtt-threshold 20000
-> saa saa2 description “saa for mac-ping” interval 500 jitter-threshold 10000
-> saa saa3 description "saa for eth-lb" interval 160
-> saa saa4 description "saa for eth-dmm" interval 300
```

2 Configure SAA “saa1” for IP ping using the **saa type ip-ping** command. For example:

```
-> saa saa1 type ip-ping destination-ip 123.22.45.66 source-ip 123.35.42.125
type-of-service 5 inter-pkt-delay 1000 num-pkts 8 payload-size 1000
```

3 Configure SAA “saa2” for MAC ping using the **saa type mac-ping** command. For example:

```
-> saa saa2 type mac-ping destination-macaddress 00:11:11:11:11:11 vlan 10
```

4 Configure SAA saa3 for Ethoam loopback using the **saa type ethoam-loopback** command. For example:

```
-> saa saa3 type ethoam-loopback target-endpoint 10 source endpoint 2 domain md1
association ma1 inter-pkt-delay 500
```

5 Configure SAA “saa4” for ETH-DMM using **saa type ethoam-two-way-delay** command. For example:

```
-> saa saa4 type ethoam-two-way-delay target-endpoint 5 source endpoint 1 domain
md2 association ma2 inter-pkt-delay 1000
```

6 Start the SAA using the **saa start** command.

```
-> saa saa1 start
-> saa saa2 start at 2009-10-13,09:00:00
```

7 Stop the SAA using the **saa stop** command.

```
-> saa saa1 stop
-> saa saa2 stop at 2009-10-13,10:00:00
```

Service Assurance Agent Overview

The Service Assurance Agent (SAA) feature is used to send periodic ping or loopback tests to peers over the network. This is done using standard IP ping packets, proprietary MAC pings, and Ethernet OAM tests. It is possible to configure a large number of test sessions on the switch, with each test having the ability to send notification traps and provide a method for determining network performance.

Each SAA test can specify threshold values for jitter and round-trip-time (RTT). When SAA processes an iteration of a test session, it will compare the results against the following criteria to see if an SNMP trap should be sent. A trap with the session name is sent if:

- At least one packet is lost.
- Warning: Average RTT/Jitter crosses 90% of threshold.
- Critical: Average RTT/Jitter at or above threshold.

When an SAA is created, an owner name is assigned to the agent. This name is based on the application that generated the SAA. For example:

- CLI SAA owner name = “USER”
- OmniVista owner name = “OV”
- Shortest Path Bridging owner name = “SPB”

The SAA feature also provides the ability to periodically record the last five iterations of all SAA sessions to an XML file on the local switch. The name of the XML file and the logging time interval are configurable SAA XML parameters.

Configuring Service Assurance Agent

This section describes how to use OmniSwitch Command Line Interface (CLI) commands to configure Service Assurance Agent (SAA) on a switch. Consider the following guidelines when configuring SAA functionality:

- Creating an SAA ID is required before the SAA type is configured. This only applies to MAC ping, IP ping, and Ethernet OAM SAAs. SAA IDs for OmniVista and SPB SAAs are automatically generated by those applications.
- Any number of SAAs can be configured (MAX 127). It is recommended not to start many aggressive SAAs (having session interval <= 10). To achieve proper scheduling of all the started SAA (aggressive and relaxed) it is recommended not to start more than 50 SAAs.
- Once the configurable SAA session timer expires (or immediately when a start is done), the session is added to the end scheduler linked list and the next session is scheduled. If there are other sessions waiting for execution, the session is processed after the other sessions have finished (first-come-first-serve). Only one session can be running a test at a time. An unlimited number of sessions can be queued up on the list. Large numbers of sessions SAAs may not observe the exact interval time.
- If the destination MAC address is found on a link aggregate, the SAA traverses all paths of the link aggregate. Each test iteration sends out multiple packets. SAA will send each packet over a different link of the aggregate. This allows SAA to test all portions of the multi-path. Calculations of the delay and jitter are available on a multi-link basis. This is only available for MAC pings.
- Ensure the interval value is greater than the execution time (number of packets * inter packet delay).
- Total execution time, that is, the product of **num-pkts** and **inter-pkt-delay** (number of packets * inter-packet delay) for a SAA iteration must be less than the sum of interval and inter-packet delay.

Configuring an SAA ID

The first step in configuring an SAA is to create an SAA ID. The **saa** command is used to create the SAA ID string (up to 32 characters), along with an optional description and time interval. For example:

```
-> saa saa2 descr "two-way eth-dm" interval 160
```

The SAA time interval specifies the amount of time, in minutes, to wait between each iteration of the SAA test. By default, the SAA time interval is set to 150 minutes and the description is set to "DEFAULT".

Additional SAA parameters include setting threshold values for jitter and round-trip-time (rtt). By default, these threshold values are set to zero (disabled). Use the **saa** command with the **jitter-threshold** and **rtt-threshold** parameters to change (enable) the threshold values. For example:

```
-> saa saa1 jitter-threshold 100 rtt-threshold 500
```

Once the SAA ID is created, then the following SAA types are configurable:

- MAC address ping (see "Configuring a MAC Address Ping SAA" on page 43-6)
- IP ping (see "Configuring an IP Ping SAA" on page 43-6)
- Ethernet OAM loopback and two-way pint (see "Configuring an Ethernet OAM SAA" on page 43-6)

Configuring a MAC Address Ping SAA

L2 SAAs enhance the service level monitoring by enabling performance measurement against any L2 address within the provider network.

To configure SAA for MAC, use the **saa type mac-ping** command. For example, the following command configures "saa5" as a MAC SAA:

```
-> saa saa5 type mac-ping destination-macaddress 00:11:11:11:11:11 vlan 10 data
"asdf" num-pkts 4
```

Configuring an IP Ping SAA

IP SAAs enhance service level monitoring to become IP application-aware by measuring both end-to-end and at the IP layer. IP SAA allows performance measurement against any IP addresses in the network (for example, switch, server, PC).

SAA IP Ping can be configured across multiple VRFs. Use the **show saa vrf** command to view the IP Ping from a specific VRF or from all the VRFs across the system.

To configure SAA for IP, use the **saa type ip-ping** command. For example, the following command configures "saa1" as an IP SAA:

```
-> vrf Blue saa "saa1" type ip-ping destination-ip 123.32.45.76 source-ip
123.35.42.124 type-of-service 4
```

Configuring an Ethernet OAM SAA

The Ethernet Service OAM implementation supports the ability to perform on-demand Ethernet loopback and two-way Ethernet frame delay measurement. These mechanisms are initiated using the **ethoam loopback** and **ethoam two-way-delay** commands. When these commands are used, the loopback or delay measurement is done on a one-time, immediate basis.

An Ethernet OAM loopback (ETH-LB) SAA and two-way frame delay measurement (ETH-DMM) SAA are supported to generate traffic in a continuous, reliable, and predictable manner to support these functions.

Configuring an ETH-LB SAA

To configure an ETH-LB SAA, use the **saa type ethoam-loopback** command. For example:

```
-> saa saa1 type ethoam-loopback target-endpoint 10 source endpoint 1 domain md1
association ma1
```

In this example, "saa1" is an existing SAA ID that is configured to run ETH-LB assurance iterations. The additional command parameters apply to the specific loopback operation. Note that these parameters are similar to those specified with the **ethoam loopback** command.

Configuring a ETH-DMM SAA

To configure a ETH-DMM SAA, use the **saa type ethoam-two-way-delay** command. For example:

```
-> saa saa2 type ethoam-two-way-delay target-endpoint 10 source endpoint 1
domain md1 association ma1
```

In this example, "saa2" is an existing SAA ID that is configured to run two-way ETH-DMM assurance test iterations. The additional command parameters apply to the specific delay measurement operation. Note that these parameters are similar to those specified with the **ethoam two-way-delay** command.

Starting and Stopping SAAs

Once an SAA ID is created and the type of SAA is configured, the SAA start and stop parameters are defined using the **saa start** and **saa stop** commands. For example:

```
-> saa saa1 start
-> saa saa1 stop
```

Both commands provide the ability to define a specific start and stop time for the SAA. For example:

```
-> saa saa2 start at 2010-09-12,09:00:00
-> saa saa2 stop at 2010-09-19,09:00:00
```

In addition, the **saa stop** command provides a **never** parameter to specify that the SAA will not stop unless a specific date and time is specified with the **saa stop** command. For example:

1 -> saa saa2 start

2 -> saa saa2 stop never

3 -> saa saa2 stop (*SAA does not stop*)

4 -> saa saa2 stop at 2010-09-19,09:00:00 (*SAA stops*)

In this example, the first command starts “saa2”. Note that because a date and time was not specified, the SAA starts immediately. The second command specifies that “saa2” will never stop unless a date and time is specified. As a result, the third command will fail because it does not specify a date and time. The fourth command, however, will successfully stop the SAA at the specified date and time.

Enabling Jitter Calculation in SAAs

Jitter is the variation in latency as measured in the variability over time of the packet latency across a network. A network with constant latency has no variation or jitter. By default, the inter-arrival jitter calculation is based on the round-trip time difference between two successive packets. The enhanced mode is to calculate inter-arrival jitter based on the formula specified in RFC 1889.

To set the jitter calculation mode to enhanced mode, use the **saa jitter-calculation enhanced** command.

```
-> saa jitter-calculation enhanced
```

Use the **no** form of this command to set the jitter calculation to default.

```
-> no saa jitter-calculation enhanced
```

To view the mode which is set for jitter-calculation, use the **show saa** command.

Configuring SAA SPB Session Parameters

The Shortest Path Bridging (SPB) feature dynamically discovers SPB-enabled switches. Each discovered switch is identified by the pairing of an SPB VLAN (BVLAN) and the backbone MAC address (BMAC) for the switch. SPB advertises these BVLAN-BMAC pairs to the SAA feature, which in turn creates and starts MAC ping sessions based on the parameters configured with this command.

Configuring an SPB SAA differs from configuring other SAA types in that an existing SAA ID is not required, because this agent is dynamically generated through SAA and SPB interaction. In this case, only the parameters that apply to dynamically created SPB SAAs are configurable using the **saa spb** command. For example:

```
-> saa spb auto-create auto-start jitter-threshold 100 rtt-threshold 500
```

In this example, parameters are configured to allow the switch to automatically create and start SPB SAA sessions with the specified jitter and round-trip-time thresholds. The default values are applied for other configurable parameters not specified in this command (for example, interval time). See the **saa saa spb** command page in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information.

To reset all SPB SAA session parameters back to their default values, use the **saa spb reset** command. For example:

```
-> saa spb reset
```

To clear all SPB SAA sessions and let the switch rebuild sessions based on the BVAN-BMAC information received from SPB, use the **saa spb flush** command. For example:

```
-> saa spb flush
```

Note that the **saa spb flush** command does not change any of the SPB SAA session parameter values.

Use the **show saa spb** command to display the current SPB SAA parameter settings. To display session information for SPB SAA, use the **show saa** command.

Generating an SAA XML History File

To configure SAA to log session information into an XML file, use the **saa xml** command with the **admin-state enable** option. For example:

```
-> saa xml interval 60 admin-state enable
```

When XML file generation is enabled, the default filename for the XML file is “saa.xml” and SAA session information is logged to the file every 20 minutes. To change the name of the file and/or the log time interval, use the **saa xml** command with the **interval** and **file-name** parameters. For example:

```
-> saa xml file-name switch1_saa.xml interval 120
-> saa xml interval 60
-> saa xml file-name edge_saa.xml
-> saa xml file-name edge2_saa.xml interval 120 admin-state enable
```

SAA will keep five iterations of all SAA sessions on the XML file. The XML file is located in the **/flash/network/** directory on the switch. The following information is logged when SAA XML file generation is enabled:

- SAA name and ID
- VRF name
- Iteration number
- Last run time
- Reason
- Packets sent/Received
- InterPktDelay
- EgressPort
- RTT min/avg/max (value is '-1' in case of failed test)
- Jitter min/avg/max (value is '-1' in case of failed test)
- Subports

Use the **show saa xml** command to display the status of XML history file generation, along with the XML filename and time interval.

Sample XML History File


```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<SystemName>ST02-CORE-01</SystemName>
     <saaId id="1" saaOwner="SPB" saaName="SPB-4007-e8-e7-32-43-01-45"
vrfName="default">
           <index id="1">
                <lastRunTime>2019-09-13,00:43:03.0</lastRunTime>
                <reason>failed</reason>
                <pktsSent>2</pktsSent>
                <pktsRcvd>0</pktsRcvd>
                <interPktDelay>100</interPktDelay>
                <egressPort>0/3</egressPort>
                <rtt>
                     <min>-1</min>
                     <avg>-1</avg>
                     <max>-1</max>
                </rtt>
                <jitter>
                     <min>-1</min>
                     <avg>-1</avg>
                     <max>-1</max>
                </jitter>
                <subport>
                     <chassis>1</chassis>
                     <egressPort>1/5/3</egressPort>
                     <pktsSent>1</pktsSent>
                     <pktsRcvd>0</pktsRcvd>
                     <rtt>
                           <min>-1</min>
                           <avg>-1</avg>
                           <max>-1</max>
                     </rtt>
                     <jitter>
                           <min>-1</min>
```

```
                        <avg>-1</avg>
                        <max>-1</max>
                    </jitter>
                </subport>
                <subport>
                    <chassis>2</chassis>
                    <egressPort>2/5/3</egressPort>
                    <pktsSent>1</pktsSent>
                    <pktsRcvd>0</pktsRcvd>
                    <rtt>
                        <min>-1</min>
                        <avg>-1</avg>
                        <max>-1</max>
                    </rtt>
                    <jitter>
                        <min>-1</min>
                        <avg>-1</avg>
                        <max>-1</max>
                    </jitter>
                </subport>
            </index>
        </saaId>
    </root>
            <index id="2">
                <lastRunTime>2019-09-13,00:41:05.0</lastRunTime>
                <reason>failed</reason>
                <pktsSent>2</pktsSent>
                <pktsRcvd>0</pktsRcvd>
                <interPktDelay>100</interPktDelay>
                <egressPort>0/3</egressPort>
                <rtt>
                    <min>-1</min>
                    <avg>-1</avg>
                    <max>-1</max>
                </rtt>
                <jitter>
                    <min>-1</min>
                    <avg>-1</avg>
                    <max>-1</max>
                </jitter>
                <subport>
                    <chassis>1</chassis>
                    <egressPort>1/5/3</egressPort>
                    <pktsSent>1</pktsSent>
                    <pktsRcvd>0</pktsRcvd>
                    <rtt>
                        <min>-1</min>
                        <avg>-1</avg>
                        <max>-1</max>
                    </rtt>
                    <jitter>
                        <min>-1</min>
                        <avg>-1</avg>
                        <max>-1</max>
                    </jitter>
                </subport>
                <subport>
                    <chassis>2</chassis>
                    <egressPort>2/5/3</egressPort>
```

```
                <pktsSent>1</pktsSent>
                <pktsRcvd>0</pktsRcvd>
                <rtt>
                      <min>-1</min>
                      <avg>-1</avg>
                      <max>-1</max>
                </rtt>
                <jitter>
                      <min>-1</min>
                      <avg>>-1</avg>
                      <max>-1</max>
                </jitter>
          </subport>
     </index>
   </saaId>
 </root>
```

Verifying the SAA Configuration

To display information about SAA on the switch, use the show commands listed below:

show saa	Displays generic configuration parameters for all the configured SAAs.
show saa type config	Displays configured SAAs for the given type.
show saa spb	Displays session parameters applies to SPB SAAs.
show saa xml	Displays configuration information for the SAA XML history file.
show saa statistics	Displays latest record, aggregated record or history.

A Software License and Copyright Statements

This appendix contains ALE USA, Inc. and third-party software vendor license and copyright statements.

ALE USA, Inc. License Agreement

ALE USA, INC. SOFTWARE LICENSE AGREEMENT

IMPORTANT. Please read the terms and conditions of this license agreement carefully before opening this package.

By opening this package, you accept and agree to the terms of this license agreement. If you are not willing to be bound by the terms of this license agreement, do not open this package. Please promptly return the product and any materials in unopened form to the place where you obtained it for a full refund.

1. **License Grant.** This is a license, not a sales agreement, between you (the "Licensee") and ALE USA, Inc. ALE USA, Inc. hereby grants to Licensee, and Licensee accepts, a non-exclusive license to use program media and computer software contained therein (the "Licensed Files") and the accompanying user documentation (collectively the "Licensed Materials"), only as authorized in this License Agreement. Licensee, subject to the terms of this License Agreement, may use one copy of the Licensed Files on the Licensee's system. Licensee agrees not to assign, sublicense, transfer, pledge, lease, rent, or share their rights under this License Agreement. Licensee may retain the program media for backup purposes with retention of the copyright and other proprietary notices. Except as authorized under this paragraph, no copies of the Licensed Materials or any portions thereof may be made by Licensee and Licensee shall not modify, decompile, disassemble, reverse engineer, or otherwise attempt to derive the Source Code. Licensee is also advised that ALE USA, Inc. products contain embedded software known as firmware which resides in silicon. Licensee may not copy the firmware or transfer the firmware to another medium.

2. **ALE USA, Inc.'s Rights.** Licensee acknowledges and agrees that the Licensed Materials are the sole property of ALE USA, Inc. and its licensors (herein "its licensors"), protected by U.S. copyright law, trademark law, and are licensed on a right to use basis. Licensee further acknowledges and agrees that all rights, title, and interest in and to the Licensed Materials are and shall remain with ALE USA, Inc. and its licensors and that no such right, license, or interest shall be asserted with respect to such copyrights and trademarks. This License Agreement does not convey to Licensee an interest in or to the Licensed Materials, but only a limited right to use revocable in accordance with the terms of this License Agreement.

3. **Confidentiality.** ALE USA, Inc. considers the Licensed Files to contain valuable trade secrets of ALE USA, Inc., the unauthorized disclosure of which could cause irreparable harm to ALE USA, Inc. Except as expressly set forth herein, Licensee agrees to use reasonable efforts not to disclose the Licensed Files to any third party and not to use the Licensed Files other than for the purpose authorized by this License Agreement. This confidentiality obligation shall continue after any termination of this License Agreement.

4. **Indemnity.** Licensee agrees to indemnify, defend and hold ALE USA, Inc. harmless from any claim, lawsuit, legal proceeding, settlement or judgment (including without limitation ALE USA, Inc.'s reasonable United States and local attorneys' and expert witnesses' fees and costs) arising out of or in connection with the unauthorized copying, marketing, performance or distribution of the Licensed Files.

5. **Limited Warranty.** ALE USA, Inc. warrants, for Licensee's benefit alone, that the program media shall, for a period of ninety (90) days from the date of commencement of this License Agreement (referred to as the Warranty Period), be free from defects in material and workmanship. ALE USA, Inc. further warrants, for Licensee benefit alone, that during the Warranty Period the Licensed Files shall operate substantially in accordance with the functional specifications in the User Guide. If during the Warranty Period, a defect in the Licensed Files appears, Licensee may return the Licensed Files to ALE USA, Inc. for either replacement or, if so elected by ALE USA, Inc., refund of amounts paid by Licensee under this License Agreement. EXCEPT FOR THE WARRANTIES SET FORTH ABOVE, THE LICENSED MATERIALS ARE LICENSED "AS IS" AND ALE USA, INC. AND ITS LICENSORS DISCLAIM ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING (WITHOUT LIMITATION) ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO LICENSEE. THIS WARRANTY GIVES THE LICENSEE SPECIFIC LEGAL RIGHTS. LICENSEE MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

6. **Limitation of Liability.** ALE USA, Inc.'s cumulative liability to Licensee or any other party for any loss or damages resulting from any claims, demands, or actions arising out of or relating to this License Agreement shall not exceed the license fee paid to ALE USA, Inc. for the Licensed Materials. IN NO EVENT SHALL ALE USA, INC. BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, OR EXEMPLARY DAMAGES OR LOST PROFITS, EVEN IF ALE USA, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION TO INCIDENTAL OR CONSEQUENTIAL DAMAGES MAY NOT APPLY TO LICENSEE.

7. **Export Control.** This product is subject to the jurisdiction of the United States. Licensee may not export or reexport the Licensed Files, without complying with all United States export laws and regulations, including but not limited to (i) obtaining prior authorization from the U.S. Department of Commerce if a validated export license is required, and (ii) obtaining "written assurances" from licensees, if required.

8. **Support and Maintenance.** Except as may be provided in a separate agreement between ALE USA, Inc. and Licensee, if any, ALE USA, Inc. is under no obligation to maintain or support the copies of the Licensed Files made and distributed hereunder and ALE USA, Inc. has no obligation to furnish Licensee with any further assistance, documentation or information of any nature or kind.

9. **Term.** This License Agreement is effective upon Licensee opening this package and shall continue until terminated. Licensee may terminate this License Agreement at any time by returning the Licensed Materials and all copies thereof and extracts therefrom to ALE USA, Inc. and certifying to ALE USA, Inc. in writing that all Licensed Materials and all copies thereof and extracts therefrom have been returned or erased by the memory of Licensee's computer or made non-readable. ALE USA, Inc. may terminate this License Agreement upon the breach by Licensee of any term hereof. Upon such termination by

ALE USA, Inc., Licensee agrees to return to ALE USA, Inc. or destroy the Licensed Materials and all copies and portions thereof.

10. **Governing Law.** This License Agreement shall be construed and governed in accordance with the laws of the State of California.

11. **Severability.** Should any term of this License Agreement be declared void or unenforceable by any court of competent jurisdiction, such declaration shall have no effect on the remaining terms herein.

12. **No Waiver.** The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

13. **Notes to United States Government Users.** Software and documentation are provided with restricted rights. Use, duplication or disclosure by the government is subject to (i) restrictions set forth in GSA ADP Schedule Contract with ALE USA, Inc.'s reseller(s), or (ii) restrictions set forth in subparagraph (c) (1) and (2) of 48 CFR 52.227-19, as applicable.

14.**Third Party Materials.** Licensee is notified that the Licensed Files contain third party software and materials licensed to ALE USA, Inc. by certain third party licensors. Some third party licensors are third part beneficiaries to this License Agreement with full rights of enforcement. Please refer to the section entitled "Third Party Licenses and Notices" on page -4 for the third party license and notice terms.

Third Party Licenses and Notices

Legal Notices applicable to any software distributed alone or in connection with the product to which this document pertains, are contained in files within the software itself located at: **/flash/foss**.

Also, if needed, we provide all FOSS (Free and Open Source Software) source code used in this release at the following URL: https://github.com/Alcatel-LucentEnterpriseData.

Index

Numerics

10/100/1000 ports
  defaults 1-3
802.1AB 15-1
  defaults 15-2
  verify information about 15-15
802.1Q
  enabling notification 15-8
  trusted ports 28-30
802.1x command 30-7, 30-9
802.3ad
  see dynamic link aggregation

A

aaa ldap-server command
  LDAP authentication 33-34
aaa radius-server command
  RADIUS authentication 33-13, 33-21
Access Control Lists
  see ACLs
access list 21-15
  creating 21-15
Access Loop 42-4
Access Node 42-4
Access Node Identifier 42-6
ACLs
  interaction with VRRP 25-11
  Layer 2 28-64
  Layer 3 28-66
  Layer 3 application examples 28-66
  multicast 28-68
  security features 28-68
actions
  combined with conditions 28-32, 28-33
  creating policy actions 28-46
Address Resolution Protocol
  see ARP
Alcatel Mapping Adjacency Protocol 16-1
alerts 38-4
AMAP
  see Alcatel Mapping Adjacency Protocol
Application example
  Learned Port Security Configuration 35-3
application example
  Ethernet OAM 39-8, 43-3
  VLAN Stacking 17-2, 17-39
application examples
  authentication servers 33-4
  Configuring 802.1AB 15-3
  DHCP Relay 23-3, 23-6, 23-7
  dynamic link aggregation 10-3, 10-25
  high availability VLANs 5-3
  ICMP policies 28-79
  IP 17-3
  IPMS 27-45, 27-47
  IPv6 19-3
  Layer 3 ACLs 28-66
  policies 28-75
  policy map groups 28-61
  Port Mapping 34-3, 34-7
  port mirroring 36-3
  port monitoring 36-4, 36-6
  QoS 28-43, 28-75
  RIP 21-3
  RMON 36-8
  Server Load Balancing 24-2, 26-3
  Spanning Tree Algorithm and Protocol 6-10, 6-43
  static link aggregation 9-3, 9-10
  switch health 36-9
  switch logging 38-2
  UDLD 2-3
  VLANs 4-3, 4-11
  VRRP 25-4, 25-26, 25-29
  VRRP3 25-30
applied configuration 28-72
  how to verify 28-74
ARP
  clearing the ARP cache 17-15
  creating a permanent entry 17-14
  deleting a permanent entry 17-14
  dynamic entry 17-14
  filtering 17-17
  local proxy 17-15
arp command 17-14
arp filter command 17-17
assigning ports to VLANs 4-6
Authenticated Switch Access
  LDAP VSAs 33-29
authentication servers
  application example 33-4
  defaults 33-2
  how backups work 33-5
  see LDAP authentication servers, RADIUS authentication servers

B

backbone VLAN 7-32, 7-42
backup router
  VRRP 25-6
boundary port 6-18
BPDU
  see Bridge Protocol Data Units
bridge 1x1 slot/port path cost command 6-37
bridge auto-vlan-containment command 6-32
bridge cist hello time command 6-28, 6-30
bridge forward delay command 6-30
bridge hello time command 6-29
bridge max age command 6-29, 6-31

bridge mode command 6-20
bridge msti priority command 6-28
bridge path cost mode command 6-31
bridge priority command 6-28
Bridge Protocol Data Units
contents 6-7
bridge slot/port command 6-31
bridge slot/port connection command 6-40
bridge slot/port priority command 6-36
built-in port groups
used with Policy Based Routing 28-80
BVLAN see backbone VLAN

C

Circuit Identifier 42-6
clear arp filter command 17-17
clear arp-cache command 17-15
Client 42-5
condition groups
for ACLs 28-54
MAC groups 28-58
network groups 28-55
port groups 28-59
sample configuration 28-54
service groups 28-57
verify information about 28-60
conditions
combined with actions 28-32, 28-33
configuring 28-45
how to create 28-45
see also condition groups
valid combinations 28-31
Configuring 802.1AB
application examples 15-3

D

debug messages 38-4
debug qos command 28-39
default route
IP 17-12
defaults
10/100/1000 ports 1-3
802.1AB 15-2
authentication servers 33-2
DHCP Relay 23-2
dynamic link aggregation 10-2, 11-2
Ethernet OAM 16-2, 39-2, 43-2
Ethernet ports 1-2, 1-3, 42-1
high availability VLANs 5-2
IP 17-3
IPMS 27-2, 27-3
IPv6 19-2
Learned Port Security 35-1
Multiple Spanning Tree 6-4
OSPF 20-2, 22-2
policy servers 29-2
Port Mapping 34-2
port mirroring 36-3
port monitoring 36-4, 36-6
QoS 28-34
RIP 21-2
RMON 36-8
RRSTP 6-5
Server Load Balancing 26-2
source learning 3-2
Spanning Tree Bridge 6-3, 12-2
Spanning Tree Port 6-3
static link aggregation 9-2
switch health 36-9
switch logging 38-2
UDLD 2-2
VLANs 4-2
VRRP 25-2
Denial of Service
see DoS
DHCP 23-5
DHCP Relay 23-1
application examples 23-3, 23-6, 23-7
defaults 23-2
forward delay time 23-9
maximum number of hops 23-10, 23-34
directed broadcast 17-26
DoS 17-27
enabling traps 17-30
setting decay value 17-30
setting penalty values 17-30
Setting Port Scan Penalty Value 17-30
DVMRP 27-6
dynamic link aggregation 10-1, 11-1
application examples 10-3, 10-25
defaults 10-2, 11-2
group actor administrative key 10-13
group actor system ID 10-14
group actor system priority 10-13
group administrative state 10-12
group partner administrative key 10-14
group partner system ID 10-15
group partner system priority 10-15
groups 10-8
assigning ports 10-9
creating groups 10-8
deleting groups 10-9
group names 10-12
removing ports 10-10
LACPDU bit settings 10-16, 10-20
LACPDU frames 10-16, 10-20
Link Aggregation Control Protocol (LACP) 10-5
MAC address 10-14, 10-15, 10-17, 10-22
port actor administrative priority 10-18
port actor port priority 10-19
port actor system administrative states 10-16
port actor system ID 10-17
port partner administrative key 10-21
port partner administrative priority 10-23
port partner administrative state 10-20
port partner administrative system ID 10-22
port partner administrative system priority 10-22

port partner port administrative status 10-23
ports 10-9
verify information about 10-28, 11-12
dynamic log
LDAP accounting servers 33-32

E

errors 38-4
Ethernet
defaults 1-2, 1-3, 42-1
flood rate 1-6
frame size 1-5
full duplex 1-4
half duplex 1-4
Ethernet OAM
application example 39-8, 43-3
configuration 39-9, 43-5
Connectivity Fault Management
Continuity Check Messages 39-5
Link Trace Messages 39-5
Loop-back Messages 39-5
defaults 16-2, 39-2, 43-2
overview 39-3
verification 16-16, 39-17
ethoam association ccm-interval command 39-10
ethoam association command 39-8
ethoam association mhf command 39-10, 39-11, 39-12
ethoam association-default command 39-10
ethoam domain command 16-4, 39-8
ethoam end-point command 39-8
ethoam intermediate-point command 39-8
ethoam linktrace command 39-12
ethoam loopback command 39-12

F

Fast Spanning Tree 6-5
filtering lists
see ACLs
flow command 1-7

H

health interval command 36-43, 39-12, 39-13
health threshold command 36-41
health threshold limits
displaying 36-42
high availability VLANs 5-1
adding egress ports 5-7, 5-8
adding ingress ports 5-7
application examples 5-3
configuration steps 5-6
creating high availability VLANs 5-6
defaults 5-2
deleting egress ports 5-7
deleting high availability VLANs 5-7
displaying 5-16
traffic flow 5-5

I

ICMP 17-33
control 17-35
QoS policies for 28-79
statistics 17-35
icmp messages command 17-34
icmp type command 17-33, 17-34
IGMP
multicast ACLs 28-64, 28-68
IGMP Spoofing 27-25
interfaces admin command 1-4
interfaces alias command 1-5
interfaces autoneg command 1-3
interfaces crossover command 1-3
interfaces duplex command 1-4
interfaces flood multicast command 1-6
interfaces max frame command 1-5
interfaces no l2 statistics command 1-4
interfaces speed command 1-3
Intermediate Agent 42-1
Internet Control Message Protocol
see ICMP
IP 17-1, 18-1
application examples 17-3, 18-2
ARP 17-13
defaults 17-3
directed broadcast 17-26
ICMP 17-33
ping 17-35
protocols 17-4, 18-10
router ID 17-19
router port 17-7
router primary address 17-19
static route 17-11, 19-19
tracing an IP route 17-36
TTL value 17-20
tunneling 17-38
UDP 17-37
verify information about 17-40, 18-17
ip access-list address command 21-15
ip access-list command 21-15
ip default-ttl command 17-20
ip directed-broadcast command 17-26
ip dos scan close-port-penalty command 17-30
ip dos scan decay command 17-30
ip dos scan tcp open-port-penalty command 17-30
ip dos scan threshold command 17-30
ip dos scan udp open-port-penalty command 17-30
ip dos trap command 17-31
ip helper boot-up command 23-13
ip helper forward delay command 23-10
ip helper maximum hops command 23-10, 23-34
ip interface command 21-3
ip load rip command 21-3, 21-6
ip multicast igmp-proxy-version command 27-13, 27-32
ip multicast neighbor-timeout command 27-11, 27-12, 27-20, 27-21, 27-22, 27-23, 27-31, 27-32, 27-38, 27-40
ip multicast query-interval command 27-18, 27-20, 27-36

ip multicast static-member command 27-15
ip multicast static-neighbor command 27-33
ip multicast static-querier command 27-14
IP Multicast Switching
see IPMS
ip multicast switching command 27-10, 27-25, 27-30, 27-42
IP multinetting 17-6
ip redist command 21-12
ip rip force-holddowntimer command 21-9
ip rip garbage-timer command 21-10
ip rip holddown-timer command 21-10
ip rip host-route command 21-11
ip rip interface auth-key command 21-18
ip rip interface auth-type command 21-18
ip rip interface command 21-3, 21-7
ip rip interface metric command 21-8
ip rip interface recv-version command 21-8
ip rip interface send-version command 21-7
ip rip interface status command 21-3, 21-7
ip rip invalid-timer command 21-10
ip rip route-tag command 21-9
ip rip status command 21-3, 21-7
ip rip update-interval command 21-9
ip route-pref command 17-19
IP router ports 17-7
modifying 17-8
removing 17-8, 18-16
ip router primary-address command 17-19
ip router router-id command 17-19
ip service command 17-31, 17-42, 17-43, 19-29, 19-30
ip slb admin command 26-3, 26-10
ip slb cluster admin status command 26-16
ip slb cluster command 26-3, 26-4, 26-11
ip slb cluster ping period command 26-14
ip slb cluster ping retries command 26-15
ip slb cluster ping timeout command 26-14
ip slb probe command 26-17, 26-18
ip slb probe expect command 26-20
ip slb probe password command 26-19
ip slb probe period command 26-18
ip slb probe port command 26-19
ip slb probe retries command 26-19
ip slb probe send command 26-20
ip slb probe status command 26-20
ip slb probe timeout command 26-18
ip slb probe url command 26-19
ip slb probe username command 26-19
ip slb server ip cluster command 26-3, 26-4, 26-13, 26-15, 26-16
ip static-route command 17-11, 19-19
IPMS 27-1
adding static members 27-15, 27-16, 27-17, 27-35
adding static queriers 27-14, 27-34
application examples 27-45, 27-47
defaults 27-2, 27-3
deleting static members 27-15, 27-35
deleting static neighbors 27-14, 27-33
deleting static queriers 27-15, 27-34
displaying 27-49, 27-50
DVMRP 27-6
enabling 27-10, 27-25, 27-26, 27-27, 27-42, 27-43, 27-44
IGMPv2 27-13, 27-33
IGMPv3 27-6, 27-13, 27-32
neighbor timeout 27-21, 27-22, 27-24, 27-39, 27-40, 27-41
optional multicast routing software 27-5
overview 27-4
PIM-SM 27-6
query interval 27-18, 27-19, 27-20, 27-36, 27-37
IPv6 19-1
addressing 19-5
application examples 19-3
autoconfiguration of addresses 19-7
defaults 19-2
tunneling types 19-18
verify information about 19-37
ipv6 access-list address command 21-15
ipv6 access-list command 21-15
ipv6 address command 19-3, 19-16
ipv6 interface command 19-3, 19-13, 19-14
ipv6 interface tunnel source destination command 19-13
ipv6 load rip command 19-3
ipv6 rip interface command 19-3
ipv6 route-pref command 19-21
ISIS-SPB 7-8

L

LACP
see dynamic link aggregation
lacp agg actor admin key command 10-3, 10-9
lacp agg actor admin state command 10-16
lacp agg actor port priority command 10-19
lacp agg actor system id command 10-17
lacp agg actor system priority command 10-18
lacp agg partner admin key command 10-21
lacp agg partner admin port command 10-23
lacp agg partner admin port priority command 10-23
lacp agg partner admin state command 10-20
lacp agg partner admin system id command 10-22
lacp agg partner admin system priority command 10-22
lacp linkagg actor admin key command 10-13
lacp linkagg actor system id command 10-14
lacp linkagg actor system priority command 10-13
lacp linkagg admin state command 10-12
lacp linkagg name command 10-12
lacp linkagg partner admin key command 10-14
lacp linkagg partner system id command 10-15
lacp linkagg partner system priority command 10-15
lacp linkagg size command 10-3, 10-8
Layer 2
statistics counters 1-4
LDAP accounting servers
dynamic log 33-32
standard attributes 33-31
LDAP authentication servers
directory entries 33-25
functional privileges 33-30
passwords for 33-28

schema extensions 33-25
SNMP attributes on authentication servers 33-30
SSL 33-35
VSAs for Authenticated Switch Access 33-29
LDAP servers
see policy servers
used for QoS policies 29-3
Learned Port Security
database table 35-9
defaults 35-1
disabling 35-12
enabling 35-12
overview 35-5
Learned Port Security Configuration
Application example 35-3
Lightweight Directory Access Protocol
see LDAP servers
line speed 1-3
link aggregation
dynamic link aggregation 10-1, 11-1
Spanning Tree parameters 6-35, 6-36, 6-38, 6-39, 6-41
static link aggregation 9-1
lldp lldpdu command 15-3
lldp notification command 15-3
lldp tlv dot1 command 15-9
lldp tlv dot3 command 15-9
lldp tlv management command 15-3
lldp tlv med command 15-10, 15-12
logged events
detail level 28-40
types of events 28-39

M

MAC address table 3-1, 3-3
aging time 3-7
duplicate MAC addresses 3-3
learned MAC addresses 3-3
static MAC addresses 3-3
MAC addresses
aging time 3-7, 6-30
dynamic link aggregation 10-14, 10-15, 10-17, 10-22
learned 3-3
statically assigned 3-3
mac-address-table command 3-3
map groups 28-61
application 28-79
creating 28-62
verifying information 28-63
master router
VRRP 25-6
MLD Zapping 27-43
MST 6-12
Internal Spanning Tree (IST) Instance 6-17
Interoperability 6-18
Migration 6-18, 6-19
MSTI 6-15
MSTP 6-12
Multiple Spanning Tree Region 6-16
Multicast Listener Discovery (MLD) 27-32
Multiple Spanning Tree
defaults 6-4

N

non combo ports
configuring 1-3

O

OSPF 21-4
defaults 20-2, 22-2
loading software 22-14
OSPF redistribution policies
deleting 17-22, 17-24, 19-24, 19-26, 21-16

P

PBB *see* Provider Backbone Bridge
pending configuration 28-72
pending policies
deleting 28-72
PIM-SM 27-6
ping
IP 17-35
ping command 17-35
policies
application examples 28-75
applied 28-72
built-in 28-37
conditions 28-45
creating policy actions 28-46
how the switch uses them 28-29
Policy Based Routing 28-80
precedence 28-49
redirect linkagg 28-77
redirect port 28-77
rules 28-47
verify information about 28-53
policies configured via PolicyView 28-73
policy action 802.1p command 28-10
policy action command 28-43
policy action map command 28-61
policy action mirror command 28-78
policy action redirect linkagg command 28-77
policy action redirect port command 28-77
policy actions
see actions
Policy Based Routing 28-80
policy condition command 28-43
policy conditions
see conditions
policy MAC groups 28-58
policy map group command 28-61
policy map groups
application example 28-61
policy network group command 28-54
policy network groups 28-55
switch default group 28-37, 28-55

policy port group command 28-54
policy port groups 28-59
policy rule command 28-43
policy server command 29-2, 29-4
policy server flush command 29-6
  compared to qos flush command 29-7
policy server load command 29-6
policy servers
  defaults 29-2
  downloading policies 29-6
  installing 29-3
  SSL 29-6
policy service group command 28-54
policy service groups 28-57
policy services 28-56
PolicyView
  LDAP policy servers 29-1
Port Mapping 34-1
  application examples 34-3, 34-7
  defaults 34-2
port mapping command 34-3
Port Mapping Session
  creating and deleting 34-4
  enabling and disabling 34-5
port mirroring 36-10
  application examples 36-3
  defaults 36-3
  direction 36-16
  disabling mirroring status 36-15
  displaying status 36-17
  enabling or disabling mirroring status 36-15
  N-to-1 port mirroring 36-14
  unblocking ports 36-15
port mirroring command 36-17, 36-22
port mirroring session
  creating 36-14
  deleting 36-17
  enabling/disabling 36-17
port mirroring source command 36-4
port mirroring source destination command 36-14, 36-16
port monitoring
  application examples 36-4, 36-6
  configuring 36-23, 36-29, 36-30
  creating a data file 36-25
  defaults 36-4, 36-6
  deleting a session 36-24, 36-32
  direction 36-26
  disabling a session 36-24
  displaying status and data 36-27, 36-30, 36-31
  enabling a session 36-24
  file overwriting 36-26
  file size 36-25
  overview 36-23, 36-28
  pausing a session 36-25
  resuming a session 36-25
  session persistence 36-25
port monitoring command 36-24, 36-25
port monitoring source command 36-23, 36-24, 36-25, 36-26, 36-29
ports
  displaying QoS information about 28-9
  Spanning Tree parameters 6-33
  VLAN assignment 4-6
port-security command 35-11
port-security learning-window command 35-12
PPPoE Intermediate Agent 42-1
Precedence
  Configured rule order 28-49
  Precedence value 28-49
precedence
  for policies 28-49
Provider Backbone Bridge
  802.1AH 7-5
  network 7-5
  SPB services 7-11

Q

QoS
  application examples 28-43, 28-75
  ASCII-file-only syntax 18-8, 28-44
  configuration overview 28-38
  defaults 28-34
  enabled/disabled 28-39
  interaction with other features 28-30
  overview 28-3
  quick steps for creating policies 28-43
  Server Load Balancing 26-12
  traffic prioritization 28-76
qos apply command 28-72
  global configuration 28-72
  policy and port configuration 28-72
qos clear log command 28-42
qos command 28-39
qos flush command 28-73
  compared to policy server flush command 29-7
qos forward log command 28-40
QoS log
  cleared 28-42
  displayed 28-41
  number of display lines 28-40
  see also logged events
qos log level command 28-39, 28-40
qos log lines command 28-40
qos port default 802.1p command 28-9
qos port default dscp command 28-9
qos port trusted command 28-9
qos qsp dcb import command 28-14
qos qsp dcb tc command 28-14
qos reset command 28-42
qos revert command 28-72
qos stats interval command 28-42
qos trust ports command 28-9
qos user-port command 28-69
Quality of Service
  see QoS

R

RADIUS accounting servers
  standard attributes 33-8
RADIUS authentication servers 33-7
  standard attributes 33-7
  VSAs 30-155, 33-10
Rapid Spanning Tree Algorithm and Protocol
  see RSTP
Redirection Policies 28-77
Remote Authentication Dial-In User Service
  see RADIUS authentication servers
Remote Identifier 42-7
resource threshold limits
  configuring 36-41
Ring Rapid Spanning Tree Algorithm and Protocol
  see RRSTP
RIP 21-1
  application examples 21-3
  defaults 21-2
  enabling 21-7
  forced hold-down timer 21-9
  garbage timer 21-10
  hold-down timer 21-10
  host route 21-11
  interface 21-7
  invalid timer 21-10
  IP 21-4
  loading 21-6
  redistribution 21-12
  security 21-18
  unloading 21-6
  update interval 21-9
  verification 21-19
  verify information about 21-19
RIP interface
  creating 21-7
  deleting 21-7
  enabling 21-7
  metric 21-8
  password 21-18
  receive option 21-8
  route tag 21-9
  send option 21-7
RMON
  application examples 36-8
  defaults 36-8
RMON events
  displaying list 36-38
  displaying specific 36-39
RMON probes
  displaying list 36-36
  displaying statistics 36-37
  enabling/disabling 36-35
rmon probes command 36-35
RMON tables
  displaying 36-36
round robin distribution algorithm
  see weighted round robin distribution algorithm
route map
  creating 21-13
  deleting 21-13
  enabling/disabling administrative status 21-16
  redistribution 21-15
  sequencing 21-14
router ID 17-19, 19-21
router port
  IP 17-7
router primary address 17-19
Routing Information Protocol
  see RIP
RRSTP 6-43
  defaults 6-5
RSTP 6-5
  port connection types 6-40
rules
  see policies

S

sampling intervals
  configuring 36-43, 39-12, 39-13
  viewing 36-43
Secure Socket Layer
  see SSL
Security Violation Mode 35-19
  restrict mode 35-19
server clusters 26-11, 26-16
server distribution algorithms 26-8
Server Load Balancing 26-1
  adding servers 26-13
  application examples 24-2, 26-3
  clusters 26-11, 26-16
  configuration steps 26-10
  defaults 26-2
  deleting clusters 26-13
  deleting servers 26-13
  disabling 26-10
  disabling clusters 26-16
  disabling servers 26-17
  displaying 26-21
  distribution algorithms 26-8
  enabling 26-10
  enabling clusters 26-16
  enabling servers 26-16
  ping period 26-14
  ping retries 26-15
  ping timeout 26-14
  QoS 26-12
  relative server weight 26-15
  server health monitoring 26-9
  servers 26-13, 26-16
  weighted round robin distribution algorithm 26-8
Server Load Balancing probes 26-17
  clusters 26-18
  configuring 26-17
  deleting 26-17
  expected status 26-20

modifying 26-18
password 26-19
probe expect 26-20
probe send 26-20
retries 26-19
servers 26-18
TCP/UDP port 26-19
timeout 26-18
URL 26-19
user name 26-19
service access command 7-56
service access l2profile command 7-59
service access vlan-xlation command 7-54, 7-57
service admin-state command 7-54
service l2profile command 7-57, 30-53
service rfp local-endpoint command 7-66
service rfp remote-endpoint command 7-67
service sap admin-state command 7-62
service sap command 7-60
service sap trusted command 7-61
service spb command 7-52
service vlan-xlation command 7-53
severity level
see switch logging
Shortest Path Bridging 7-1, 7-5
benefits 7-1
bridge ID 7-48
bridge priority 7-48
BVLAN 7-42
ISIS-SPB 7-8
services 7-11
shortest path trees 7-7
SPT 7-7
system ID 7-48
topology example 7-12
show arp command 17-14
show arp filter command 17-17, 17-31
show health command 36-44
show health interval command 36-43
show health threshold command 36-9, 36-42
show icmp control command 17-35
show icmp statistics command 17-35
show ip config command 17-20, 17-27
show ip interface command 17-8
show ip redist command 21-16
show ip rip command 21-7
show ip rip interface command 21-7
show ip route command 17-12, 19-19
show ip route-map command 21-13
show ipv6 interface command 19-13
show linkagg command 9-11
show linkagg port command 9-11
show lldp remote-system command 15-3
show lldp statistics command 15-3
show log swlog command 38-9
show mac-address-table port-mac command 5-16
show policy rule command 26-12
show policy server long command 29-6
show port mirroring status command 36-17
show port monitoring file command 36-27
show port-security command 35-3
show qos log command 28-41
show rmon events command 36-36
show rmon probes command 36-8, 36-36
show service access command 7-55, 7-59
show service command 7-54
show service ports command 7-62
show service rfp command 7-69
show service rfp configuration command 7-67
show spb isis bvlans command 7-44
show spb isis interface command 7-46
show tcp ports command 17-36
show tcp statistics command 17-36
show udld configuration command 2-3
show udld statistics port command 2-3
show udp ports command 17-37
show udp statistics command 17-37
show vlan svlan command 14-13, 37-35, 40-10
show vlan svlan port-binding command 14-13, 40-10
show vlan svlan port-config command 14-13, 37-35, 40-10
SLB
see Server Load Balancing
SNMP
attributes for LDAP authentication servers 33-30
source learning 3-1
defaults 3-2
MAC address table 3-1, 3-3
source learning time limit 35-12
Spanning Tree Algorithm and Protocol 6-1, 12-1
1x1 operating mode 4-9, 6-20, 6-22
application examples 6-10, 6-43
bridge ID 6-7, 6-28
Bridge Protocol Data Units 6-7, 6-29, 6-30
bridged ports 6-33
designated bridge 6-5
flat operating mode 4-9, 6-20, 6-21
path cost 6-36, 6-37
port connection types 6-40
Port ID 6-7
port ID 6-35
port path cost 6-5
port roles 6-5
port states 6-6, 6-38
root bridge 6-5, 6-29, 6-30
root path cost 6-5
topology 6-5, 6-11
Topology Change Notification 6-8
Spanning Tree Bridge
defaults 6-3, 12-2
Spanning Tree bridge parameters
802.1D standard protocol 6-27
802.1s multiple spanning tree protocol 6-27
802.1w rapid reconfiguration protocol 6-27
automatic VLAN containment 6-31
forward delay time 6-30
hello time 6-29
maximum age time 6-29
priority 6-28

Spanning Tree Port
defaults 6-3
Spanning Tree port parameters 6-33
connection type 6-40
link aggregate ports 6-35, 6-36, 6-38, 6-39, 6-41
mode 6-38
path cost 6-36
priority 6-35
spb bvlan command 7-42
spb isis admin-state command 7-52
spb isis area-address command 7-48
spb isis bridge-priority command 7-48
spb isis bvlan ect-id command 7-43
spb isis bvlan tandem-multicast-mode command 7-43
spb isis control-address command 7-49
spb isis control-bvlan command 7-43
spb isis graceful-restart command 7-51
spb isis graceful-restart helper command 7-51
spb isis interface command 7-44
spb isis lsp-wait command 7-50
spb isis overload command 7-50
spb isis overload-on-boot command 7-51
spb isis source-id command 7-48
spb isis spf-wait command 7-49
SPB *see* Shortest Path Bridging
SSL
for LDAP authentication servers 33-35
policy servers 29-6
static agg agg num command 9-3, 9-7
static link aggregation 9-1
adding ports 9-7
application examples 9-3, 9-10
configuration steps 9-6
creating 9-7
defaults 9-2
deleting 9-7
deleting ports 9-8
disabling 9-9
enabling 9-9
group names 9-9
groups 9-5, 10-5
overview 9-5, 10-5
verify information about 9-11
static linkagg admin state command 9-9
static linkagg name command 9-9
static linkagg size command 9-3, 9-7
static MAC addresses 3-3
static route
IP 17-11, 19-19
metric 17-11, 19-19
subnet mask 17-11
subnet mask 17-11
switch health
application examples 36-9
defaults 36-9
monitoring 36-40
switch health statistics
resetting 36-45
viewing 36-44
switch logging
application examples 38-2
application ID 38-4
defaults 38-2
output 38-5, 38-11
severity level 38-5
swlog appid level command 38-4
swlog clear command 38-8
swlog command 38-2, 38-4
swlog output command 28-41
swlog output command 38-5, 38-11

T

TCN BPDU
see Topology Change Notification BPDU
TCP
statistics 17-36
time-to-live
see TTL
Topology Change Notification BPDU 6-8
traceroute command 17-36
tracking
VRRP 25-8
traffic prioritization 28-76
trap port link command 1-4
traps
port link messages 1-4
Trust 42-5
trusted ports
see also ports
used with QoS policies 28-10
TTL value 17-20
Tunneling 19-9

U

UDLD
application examples 2-3
defaults 2-2
disabling on port 2-6
disabling on switch 2-6
enabling on port 2-6
overview 2-4
show 2-8
udld command 2-3
udld port command 2-3
UDP 17-37
statistics 17-37
unp l2-profile command 30-54
User Datagram Protocol
see UDP
users
functional privileges 33-30

V

Vendor Specific Attributes
see VSAs
Virtual Router Redundancy Protocol

see VRRP
virtual routers 25-6
vlan 802.1q command 4-6
vlan authentication command 4-9
vlan command 5-6, 17-3, 18-2, 21-3
vlan port command
and 802.1X ports 30-160
vlan port default command 4-6, 17-3, 21-3
vlan port-mac egress-port command 5-7, 5-8
vlan port-mac ingress-port command 5-7
vlan router ip command 17-3, 18-2, 18-3
VLAN Stacking
application example 17-2, 17-39
display list of all or range of configured SVLANs 12-30
displaying the configuration 37-35
vlan stp command 4-9
vlan svlan command 6-26
VLANs 4-1, 4-5, 12-3
administrative status 4-5
application examples 4-3, 4-11
default VLAN 4-6
defaults 4-2
description 4-5
high availability VLANs 5-1
IP multinetting 17-6
IP router ports 17-7
MAC address aging time 3-7
operational status 4-4
port assignment 4-6
Spanning Tree status 4-9
VLAN ID 4-4
VRRP 25-1
ACLs 25-11
application example 25-4, 25-26, 25-29
ARP request 25-7
backup router 25-6
defaults 25-2
MAC address 25-7
master router 25-6
tracking 25-8
virtual routers 25-6
vrrp command 25-11
defaults 25-2
vrrp delay command 25-18
vrrp ip command 25-11
vrrp track command 25-23
vrrp track-association command 25-23
VRRP3
application examples 25-30
VSAs
for LDAP servers 33-29
for RADIUS authentication 33-7
setting up for RADIUS servers 30-155, 33-10

W

warnings 38-4
weighted round robin distribution algorithm 26-8