

**Part No. 060319-10, Rev. A
December 2010**

OmniSwitch AOS Release 7 Network Configuration Guide



www.alcatel-lucent.com

**This user guide documents release 7.1.1 of the OmniSwitch 10K.
The functionality described in this guide is subject to change without notice.**

Copyright © 2010 by Alcatel-Lucent. All rights reserved. This document may not be reproduced in whole or in part without the express written permission of Alcatel-Lucent.

Alcatel-Lucent® and the Alcatel-Lucent logo are registered trademarks of Alcatel-Lucent. Xylan®, OmniSwitch®, OmniStack®, and Alcatel-Lucent OmniVista® are registered trademarks of Alcatel-Lucent.

OmniAccess™, Omni Switch/Router™, PolicyView™, RouterView™, SwitchManager™, VoiceView™, WebView™, X-Cell™, X-Vision™, and the Xylan logo are trademarks of Alcatel-Lucent.

This OmniSwitch product contains components which may be covered by one or more of the following U.S. Patents:

- U.S. Patent No. 6,339,830
- U.S. Patent No. 6,070,243
- U.S. Patent No. 6,061,368
- U.S. Patent No. 5,394,402
- U.S. Patent No. 6,047,024
- U.S. Patent No. 6,314,106
- U.S. Patent No. 6,542,507
- U.S. Patent No. 6,874,090



**26801 West Agoura Road
Calabasas, CA 91301
(818) 880-3500 FAX (818) 880-3505
support@ind.alcatel.com**

**US Customer Support—(800) 995-2696
International Customer Support—(818) 878-4507
Internet—service.esd.alcatel-lucent.com**

Contents

	About This Guide	xxi
	Supported Platforms	xxi
	Who Should Read this Manual?	xxi
	When Should I Read this Manual?	xxi
	What is in this Manual?	xxii
	What is Not in this Manual?	xxii
	How is the Information Organized?	xxii
	Documentation Roadmap	xxiii
	Related Documentation	xxv
	Technical Support	xxvi
Chapter 1	Configuring Ethernet Ports	1-1
	In This Chapter	1-1
	Ethernet Specifications	1-2
	Ethernet Port Defaults	1-2
	Ethernet Ports Overview	1-3
	Valid Port Settings	1-3
	Configuring Ethernet Port Parameters	1-3
	Enabling and Disabling Autonegotiation	1-3
	Configuring Crossover Settings	1-3
	Setting Interface Line Speed	1-4
	Configuring Duplex Mode	1-4
	Setting Trap Port Link Messages	1-4
	Resetting Statistics Counters	1-4
	Enabling and Disabling Interfaces	1-5
	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	1-6
Chapter 2	Configuring UDLD	2-1
	In This Chapter	2-1
	UDLD Specifications	2-2
	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 Specifications	3-2
	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 Specifications	4-2
	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	4-6
	Modifying the VLAN Description	4-6
	Assigning Ports to VLANs	4-6
	Changing the Default VLAN Assignment for a Port	4-7
	Using 802.1Q Tagging	4-7
	Enabling/Disabling Spanning Tree for a VLAN	4-9
	Enabling/Disabling Source Learning	4-9
	Configuring VLAN Router Interfaces	4-10

	Bridging VLANs Across Multiple Switches	4-11
	Verifying the VLAN Configuration	4-13
	Understanding Port Output	4-13
Chapter 5	Configuring Spanning Tree Parameters	5-1
	In This Chapter	5-2
	Spanning Tree Specifications	5-3
	Spanning Tree Bridge Parameter Defaults	5-4
	Spanning Tree Port Parameter Defaults	5-4
	Multiple Spanning Tree (MST) Region Defaults	5-5
	Spanning Tree Overview	5-6
	How the Spanning Tree Topology is Calculated	5-6
	MST General Overview	5-12
	How MSTP Works	5-12
	Comparing MSTP with STP and RSTP	5-15
	What is a Multiple Spanning Tree Instance (MSTI)	5-15
	What is a Multiple Spanning Tree Region	5-16
	What is the Common Spanning Tree	5-17
	What is the Internal Spanning Tree (IST) Instance	5-17
	What is the Common and Internal Spanning Tree Instance	5-17
	MST Configuration Overview	5-17
	MST Interoperability and Migration	5-18
	Spanning Tree Operating Modes	5-20
	Using Flat Spanning Tree Mode	5-20
	Using Per-VLAN Spanning Tree Mode	5-21
	Using Per-VLAN Spanning Tree Mode with PVST+	5-22
	OmniSwitch PVST+ Interoperability	5-23
	Using Spanning Tree Configuration Commands	5-25
	Configuring STP Bridge Parameters	5-26
	Selecting the Spantree Protocol	5-27
	Configuring the Bridge Priority	5-27
	Configuring the Bridge Hello Time	5-28
	Configuring the Bridge Max-Age Time	5-29
	Configuring the Forward Delay Time for the Switch	5-29
	Enabling/Disabling the VLAN BPDU Switching Status	5-30
	Configuring the Path Cost Mode	5-31
	Using Automatic VLAN Containment	5-31
	Configuring STP Port Parameters	5-33
	Enabling/Disabling Spanning Tree on a Port	5-34
	Configuring Port Priority	5-36
	Configuring Port Path Cost	5-37
	Configuring Port Mode	5-40
	Configuring Port Connection Type	5-41
	Configuring the Edge Port Status	5-42
	Restricting Port Roles (Root Guard)	5-43
	Restricting TCN Propagation	5-43

	Limiting BPDU Transmission	5-43
	Sample Spanning Tree Configuration	5-44
	Example Network Overview	5-44
	Example Network Configuration Steps	5-45
	Sample MST Region Configuration	5-47
	Sample MSTI Configuration	5-49
	Verifying the Spanning Tree Configuration	5-52
Chapter 6	Configuring Static Link Aggregation	6-1
	In This Chapter	6-1
	Static Link Aggregation Specifications	6-2
	Static Link Aggregation Default Values	6-2
	Quick Steps for Configuring Static Link Aggregation	6-3
	Static Link Aggregation Overview	6-5
	Static Link Aggregation Operation	6-5
	Relationship to Other Features	6-6
	Configuring Static Link Aggregation Groups	6-6
	Configuring Mandatory Static Link Aggregate Parameters	6-6
	Creating and Deleting a Static Link Aggregate Group	6-7
	Adding and Deleting Ports in a Static Aggregate Group	6-8
	Modifying Static Aggregation Group Parameters	6-9
	Modifying the Static Aggregate Group Name	6-9
	Modifying the Static Aggregate Group Administrative State	6-9
	Application Example	6-10
	Displaying Static Link Aggregation Configuration and Statistics	6-11
Chapter 7	Configuring Dynamic Link Aggregation	7-1
	In This Chapter	7-1
	Dynamic Link Aggregation Specifications	7-2
	Dynamic Link Aggregation Default Values	7-3
	Quick Steps for Configuring Dynamic Link Aggregation	7-4
	Dynamic Link Aggregation Overview	7-6
	Dynamic Link Aggregation Operation	7-6
	Relationship to Other Features	7-8
	Configuring Dynamic Link Aggregate Groups	7-8
	Configuring Mandatory Dynamic Link Aggregate Parameters	7-9
	Creating and Deleting a Dynamic Aggregate Group	7-9
	Configuring Ports to Join and Removing Ports in a Dynamic Aggregate Group	7-10
	Modifying Dynamic Link Aggregate Group Parameters	7-12
	Modifying Dynamic Aggregate Group Parameters	7-12

	Modifying Dynamic Link Aggregate Actor Port Parameters	7-17
	Modifying Dynamic Aggregate Partner Port Parameters	7-21
	Application Examples	7-27
	Sample Network Overview	7-27
	Link Aggregation and Spanning Tree Example	7-28
	Link Aggregation and QoS Example	7-29
	Displaying Dynamic Link Aggregation Configuration and Statistics	7-30
Chapter 8	Configuring Multi-chassis Link Aggregation	8-1
	In This Chapter	8-2
	Multi-chassis Link Aggregation Specifications	8-3
	Multi-chassis Link Aggregation Default Values	8-4
	Quick Steps for Configuring MC-LAG	8-5
	MC-LAG Overview	8-9
	MC-LAG Concepts and Components	8-9
	Benefits of MC-LAG	8-10
	MC-LAG Principle	8-11
	MC-LAG Loop Detection	8-12
	MC-LAG Topologies	8-13
	Basic MC-LAG Building Block	8-13
	Recommended Topologies	8-13
	Topologies Not Recommended	8-15
	Unsupported Topologies	8-15
	MC-LAG Packet Flow	8-18
	Layer 2 Switching over MC-LAG	8-18
	Configuring MC-LAG	8-21
	MC-LAG Configuration Guidelines	8-21
	Configuring the Chassis-ID	8-23
	Configuring the IPC-VLAN	8-23
	Configuring the Hello-Interval	8-23
	Configuring Aggregate Identifier Ranges	8-23
	Creating the Virtual Fabric Link (VFL)	8-24
	Configuring MC-LAG Aggregates	8-24
	Configuring the VIP VLAN	8-24
	Recommended Configuration Parameters	8-25
	Verifying Parameter Consistency	8-25
	Interaction with Other Features	8-27
	Spanning Tree Protocol	8-27
	Ethernet Ring Protection	8-27
	Link Aggregation	8-27
	Multicast	8-27
	Source Learning	8-27
	IPv4	8-27
	MC-LAG Configuration Example	8-28
	Displaying MC-LAG Configuration and Statistics	8-30

Chapter 9	Configuring ERP	9-1
	In This Chapter	9-1
	ERP Specifications	9-2
	ERP Defaults	9-2
	ERP Overview	9-3
	How Does ERP Work?	9-4
	ERP and RRSTP Differences	9-7
	Interaction With Other Features	9-8
	Quick Steps for Configuring ERP with Standard VLANs	9-9
	Quick Steps for Configuring ERP with VLAN Stacking	9-10
	ERP Configuration Overview and Guidelines	9-11
	Configuring an ERP Ring	9-12
	Adding VLANs to Ring Ports	9-12
	Configuring an RPL Port	9-13
	Setting the Wait-to-Restore Timer	9-13
	Setting the Guard Timer	9-14
	Configuring ERP with VLAN Stacking NNIs	9-14
	Clearing ERP Statistics	9-16
	Sample Ethernet Ring Protection Configuration	9-17
	Example ERP Overview	9-17
	Example ERP Configuration Steps	9-18
	Verifying the ERP Configuration	9-19
Chapter 10	Configuring 802.1AB	10-1
	In This Chapter	10-1
	802.1AB Specifications	10-2
	802.1AB Defaults Table.	10-2
	Quick Steps for Configuring 802.1AB	10-2
	Quick Steps for Configuring LLDP-MED Network Policy	10-4
	LLDP-MED Network Policy for Fixed Ports	10-4
	802.1AB Overview	10-5
	Mandatory TLVs	10-5
	Optional TLVs	10-6
	LLDP-Media Endpoint Devices	10-7
	LLDP Agent Operation	10-9
	LLDPDU Transmission and Reception	10-9
	Aging Time	10-9
	Configuring 802.1AB	10-10
	Configuring LLDPDU Flow	10-10
	Enabling and Disabling Notification	10-10
	Enabling and Disabling Management TLV	10-10
	Enabling and Disabling 802.1 TLV	10-11
	Enabling and Disabling 802.3 TLV	10-11

	Enabling and Disabling MED TLV	10-12
	Setting the Transmit Interval	10-12
	Setting the Transmit Hold Multiplier Value	10-12
	Setting the Transmit Delay	10-12
	Setting the Transmit Fast Start Count	10-13
	Setting the Reinit Delay	10-13
	Setting the Notification Interval	10-13
	Verifying 802.1AB Configuration	10-14
Chapter 11	Configuring IP	11-1
	In This Chapter	11-1
	IP Specifications	11-3
	IP Defaults	11-3
	Quick Steps for Configuring IP Forwarding	11-4
	IP Overview	11-5
	IP Protocols	11-5
	IP Forwarding	11-7
	Configuring an IP Router Interface	11-7
	Configuring a Loopback0 Interface	11-10
	Creating a Static Route or Recursive Static Route	11-11
	Creating a Default Route	11-12
	Configuring Address Resolution Protocol (ARP)	11-12
	IP Configuration	11-16
	Configuring the Router Primary Address	11-16
	Configuring the Router ID	11-16
	Configuring the Route Preference of a Router	11-16
	Configuring the Time-to-Live (TTL) Value	11-17
	Configuring Route Map Redistribution	11-17
	IP-Directed Broadcasts	11-23
	Denial of Service (DoS) Filtering	11-23
	Enabling/Disabling IP Services	11-27
	Managing IP	11-29
	Internet Control Message Protocol (ICMP)	11-29
	Using the Ping Command	11-32
	Tracing an IP Route	11-33
	Displaying TCP Information	11-33
	Displaying UDP Information	11-33
	Tunneling	11-34
	Generic Routing Encapsulation	11-34
	IP Encapsulation within IP	11-34
	Tunneling operation	11-35
	Configuring a Tunnel Interface	11-35
	Verifying the IP Configuration	11-37

Chapter 12	Configuring Multiple VRF	12-1
	In This Chapter	12-1
	VRF Specifications	12-2
	VRF Defaults	12-2
	Quick Steps for Configuring Multiple VRF	12-3
	Multiple VRF Overview	12-6
	Using the VRF Command Line Interface	12-8
	ASCII-File-Only Syntax	12-8
	VRF Interaction With Other Features	12-9
	AAA RADIUS Servers	12-10
	BGPv4	12-10
	IP-IP and GRE Tunnels	12-10
	Management Applications (Telnet and SSH)	12-10
	Quality of Service (QoS)	12-10
	SNMP	12-11
	VLANs	12-11
	UDP/DHCP Relay	12-11
	Configuring VRF Instances	12-12
	Selecting a VRF Instance	12-13
	Assigning IP Interfaces to a VRF Instance	12-14
	Configuring Routing Protocols for a Specific VRF Instance	12-14
	Removing a VRF Instance	12-14
	Verifying the VRF Configuration	12-15
Chapter 13	Configuring IPv6	13-1
	In This Chapter	13-1
	IPv6 Specifications	13-2
	IPv6 Defaults	13-3
	Quick Steps for Configuring IPv6 Routing	13-4
	IPv6 Overview	13-5
	IPv6 Addressing	13-6
	Tunneling IPv6 over IPv4	13-10
	Configuring an IPv6 Interface	13-13
	Configuring a Unique Local IPv6 Unicast Address	13-14
	Modifying an IPv6 Interface	13-14
	Removing an IPv6 Interface	13-14
	Assigning IPv6 Addresses	13-15
	Removing an IPv6 Address	13-16
	Configuring IPv6 Tunnel Interfaces	13-17
	Creating an IPv6 Static Route	13-18
	Configuring the Route Preference of a Router	13-19

	Configuring Route Map Redistribution	13-20
	Verifying the IPv6 Configuration	13-26
Chapter 14	Configuring IPsec	14-1
	In This Chapter	14-1
	IPsec Specifications	14-2
	IPsec Defaults	14-2
	Quick Steps for Configuring an IPsec AH Policy	14-3
	Quick Steps for Configuring an IPsec Discard Policy	14-4
	IPsec Overview	14-4
	Encapsulating Security Payload (ESP)	14-5
	Authentication Header (AH)	14-6
	IPsec on the OmniSwitch	14-7
	Securing Traffic Using IPsec	14-7
	Discarding Traffic using IPsec	14-8
	Configuring IPsec on the OmniSwitch	14-9
	Configuring an IPsec Master Key	14-9
	Configuring an IPsec Policy	14-10
	Configuring an IPsec SA	14-14
	Additional Examples	14-18
	Configuring ESP	14-18
	Discarding RIPng Packets	14-20
	Verifying IPsec Configuration	14-20
Chapter 15	Configuring RIP	15-1
	In This Chapter	15-1
	RIP Specifications	15-2
	RIP Defaults	15-2
	Quick Steps for Configuring RIP Routing	15-3
	RIP Overview	15-4
	RIP Version 2	15-5
	RIP Routing	15-6
	Loading RIP	15-6
	Enabling RIP	15-7
	Creating a RIP Interface	15-7
	Enabling a RIP Interface	15-7
	RIP Options	15-9
	Configuring the RIP Forced Hold-Down Interval	15-9
	Configuring the RIP Update Interval	15-9
	Configuring the RIP Invalid Timer	15-10
	Configuring the RIP Garbage Timer	15-10
	Configuring the RIP Hold-Down Timer	15-10
	Reducing the Frequency of RIP Routing Updates	15-10
	Enabling a RIP Host Route	15-11

	Configuring Redistribution	15-12
	RIP Security	15-18
	Configuring Authentication Type	15-18
	Configuring Passwords	15-18
	Verifying the RIP Configuration	15-19
Chapter 16	Configuring BFD	16-1
	In This Chapter	16-1
	BFD Specifications	16-2
	BFD Defaults	16-2
	Quick Steps for Configuring BFD	16-3
	Quick Steps for Configuring BFD Support for Layer 3 Protocols	16-5
	BFD Overview	16-7
	Benefits of Using BFD For Failure Detection	16-8
	How the BFD Protocol Works	16-8
	Operational Mode and Echo Function	16-9
	BFD Packet Formats	16-9
	BFD Session Establishment	16-10
	Configuring BFD	16-11
	Configuring BFD Session Parameters	16-11
	Configuring BFD Support for Layer 3 Protocols	16-15
	BFD Application Example	16-23
	Example Network Overview	16-23
	Verifying the BFD Configuration	16-28
Chapter 17	Configuring DHCP Relay	17-1
	In This Chapter	17-1
	DHCP Relay Specifications	17-2
	DHCP Relay Defaults	17-3
	Quick Steps for Setting Up DHCP Relay	17-4
	DHCP Relay Overview	17-5
	DHCP	17-6
	DHCP and the OmniSwitch	17-6
	External DHCP Relay Application	17-7
	Internal DHCP Relay	17-8
	DHCP Relay Implementation	17-9
	Global DHCP	17-9
	Per-VLAN DHCP	17-9
	Configuring BOOTP/DHCP Relay Parameters	17-10
	Setting the Forward Delay	17-10
	Setting Maximum Hops	17-11
	Setting the Relay Forwarding Option	17-11

	Using Automatic IP Configuration	17-12
	Enabling Automatic IP Configuration	17-12
	Configuring UDP Port Relay	17-13
	Enabling/Disabling UDP Port Relay	17-14
	Specifying a Forwarding VLAN	17-14
	How the Relay Agent Processes DHCP Packets from the Client	17-15
	Verifying the DHCP Relay Configuration	17-17
Chapter 18	Configuring VRRP	18-1
	In This Chapter	18-1
	VRRP Specifications	18-3
	VRRP Defaults	18-3
	Quick Steps for Creating a Virtual Router	18-5
	VRRP Overview	18-6
	Why Use VRRP?	18-7
	Definition of a Virtual Router	18-7
	VRRP MAC Addresses	18-8
	VRRP Startup Delay	18-9
	VRRP Tracking	18-9
	Configuring Collective Management Functionality	18-9
	Interaction With Other Features	18-9
	VRRP Configuration Overview	18-10
	Basic Virtual Router Configuration	18-10
	Creating/Deleting a Virtual Router	18-10
	Specifying an IP Address for a Virtual Router	18-11
	Configuring the Advertisement Interval	18-12
	Configuring Virtual Router Priority	18-12
	Setting Preemption for Virtual Routers	18-12
	Enabling/Disabling a Virtual Router	18-13
	Setting VRRP Traps	18-14
	Setting VRRP Startup Delay	18-14
	Configuring Collective Management Functionality	18-14
	Verifying the VRRP Configuration	18-18
	VRRPv3 Configuration Overview	18-19
	Basic VRRPv3 Virtual Router Configuration	18-19
	Creating/Deleting a VRRPv3 Virtual Router	18-19
	Specifying an IPv6 Address for a VRRPv3 Virtual Router	18-20
	Configuring the VRRPv3 Advertisement Interval	18-21
	Configuring the VRRPv3 Virtual Router Priority	18-21
	Setting Preemption for VRRPv3 Virtual Routers	18-22
	Enabling/Disabling a VRRPv3 Virtual Router	18-23
	Setting VRRPv3 Traps	18-23
	Verifying the VRRPv3 Configuration	18-24
	Creating Tracking Policies	18-25
	Associating a Tracking Policy with a VRRPv2/VRRPv3 Virtual Router	18-25

	VRRP Application Example	18-26
	VRRP Tracking Example	18-28
	VRRPv3 Application Example	18-30
	VRRPv3 Tracking Example	18-31
Chapter 19	Configuring Server Load Balancing	19-1
	In This Chapter	19-1
	Server Load Balancing Specifications	19-2
	Server Load Balancing Default Values	19-3
	Quick Steps for Configuring Server Load Balancing	19-4
	Quick Steps for Configuring a QoS Policy Condition Cluster	19-5
	Server Load Balancing Overview	19-7
	Server Load Balancing Cluster Identification	19-7
	Server Load Balancing Example	19-8
	Weighted Round Robin Distribution Algorithm	19-9
	Server Health Monitoring	19-10
	Configuring Server Load Balancing on a Switch	19-11
	Enabling and Disabling Server Load Balancing	19-11
	Configuring and Deleting SLB Clusters	19-12
	Assigning Servers to and Removing Servers from a Cluster	19-14
	Modifying Optional Parameters	19-15
	Modifying the Ping Period	19-15
	Modifying the Ping Timeout	19-15
	Modifying the Ping Retries	19-16
	Modifying the Relative Weight of a Physical Server	19-16
	Taking Clusters and Servers On/Off Line	19-17
	Taking a Cluster On/Off Line	19-17
	Taking a Server On/Off Line	19-17
	Configuring SLB Probes	19-18
	Creating SLB Probes	19-18
	Deleting SLB Probes	19-18
	Associating a Probe with a Cluster	19-19
	Associating a Probe with a Server	19-19
	Modifying SLB Probes	19-19
	Displaying Server Load Balancing Status and Statistics	19-22
Chapter 20	Configuring IP Multicast Switching	20-1
	In This Chapter	20-1
	IPMS Specifications	20-2
	IPMSv6 Specifications	20-2
	IPMS Default Values	20-3
	IPMSv6 Default Values	20-4
	IPMS Overview	20-5

IPMS Example	20-5
Reserved IP Multicast Addresses	20-6
IP Multicast Routing	20-6
Configuring IPMS on a Switch	20-8
Enabling and Disabling IP Multicast Status	20-8
Enabling and Disabling IGMP Querier-forwarding	20-9
Configuring and Restoring the IGMP Version	20-9
Configuring and Removing an IGMP Static Neighbor	20-10
Configuring and Removing an IGMP Static Querier	20-11
Configuring and Removing an IGMP Static Group	20-11
Modifying IPMS Parameters	20-13
Modifying the IGMP Query Interval	20-13
Modifying the IGMP Last Member Query Interval	20-13
Modifying the IGMP Query Response Interval	20-14
Modifying the IGMP Router Timeout	20-15
Modifying the Source Timeout	20-16
Enabling and Disabling IGMP Querying	20-17
Modifying the IGMP Robustness Variable	20-17
Enabling and Disabling the IGMP Spoofing	20-18
Enabling and Disabling the IGMP Zapping	20-19
Limiting IGMP Multicast Groups	20-20
IPMSv6 Overview	20-21
IPMSv6 Example	20-21
Reserved IPv6 Multicast Addresses	20-22
MLD Version 2	20-22
Configuring IPMSv6 on a Switch	20-23
Enabling and Disabling IPv6 Multicast Status	20-23
Enabling and Disabling MLD Querier-forwarding	20-24
Configuring and Restoring the MLD Version	20-24
Configuring and Removing an MLD Static Neighbor	20-25
Configuring and Removing an MLD Static Querier	20-26
Configuring and Removing an MLD Static Group	20-26
Modifying IPMSv6 Parameters	20-28
Modifying the MLD Query Interval	20-28
Modifying the MLD Last Member Query Interval	20-28
Modifying the MLD Query Response Interval	20-29
Modifying the MLD Router Timeout	20-30
Modifying the Source Timeout	20-30
Enabling and Disabling the MLD Querying	20-31
Modifying the MLD Robustness Variable	20-32
Enabling and Disabling the MLD Spoofing	20-33
Enabling and Disabling the MLD Zapping	20-33
Limiting MLD Multicast Groups	20-34
IPMS Application Example	20-36
IPMSv6 Application Example	20-38
Displaying IPMS Configurations and Statistics	20-40
Displaying IPMSv6 Configurations and Statistics	20-41

Chapter 21	Configuring QoS	21-1
	In This Chapter	21-1
	QoS Specifications	21-3
	QoS General Overview	21-4
	Classification	21-6
	How Traffic is Classified and Marked	21-6
	Configuring Trusted Ports	21-8
	Congestion Management	21-10
	Queue Sets	21-11
	Queue Mapping Tables	21-16
	Congestion Avoidance	21-18
	WRED Profiles	21-18
	Traffic Policing and Shaping	21-20
	Policing	21-20
	Shaping	21-21
	Tri-Color Marking	21-21
	Configuring Policy Bandwidth Policing	21-23
	Configuring Port Bandwidth Shaping	21-25
	QoS Policy Overview	21-26
	QoS Defaults	21-30
	Global QoS Defaults	21-30
	QoS Port Defaults	21-31
	Queue Management Defaults	21-31
	Policy Rule Defaults	21-32
	Policy Action Defaults	21-32
	Default (Built-in) Policies	21-33
	Configuring QoS	21-34
	Configuring Global QoS Parameters	21-35
	Enabling/Disabling QoS	21-35
	Using the QoS Log	21-35
	Setting the Statistics Interval	21-38
	Returning the Global Configuration to Defaults	21-38
	Verifying Global Settings	21-38
	Creating Policies	21-39
	Quick Steps for Creating Policies	21-39
	ASCII-File-Only Syntax	21-40
	Creating Policy Conditions	21-41
	Creating Policy Actions	21-42
	Creating Policy Rules	21-43
	Verifying Policy Configuration	21-47
	Using Condition Groups in Policies	21-48
	Sample Group Configuration	21-48
	Creating Network Groups	21-49
	Creating Services	21-50
	Creating Service Groups	21-51

Creating MAC Groups	21-52
Creating Port Groups	21-53
Verifying Condition Group Configuration	21-55
Using Map Groups	21-56
Sample Map Group Configuration	21-56
How Map Groups Work	21-57
Creating Map Groups	21-57
Verifying Map Group Configuration	21-58
Using Access Control Lists	21-59
Layer 2 ACLs	21-59
Layer 3 ACLs	21-60
IPv6 ACLs	21-62
Multicast Filtering ACLs	21-62
Using ACL Security Features	21-63
Applying the Configuration	21-66
Interaction With LDAP Policies	21-67
Verifying the Applied Policy Configuration	21-68
Policy Applications	21-69
Basic QoS Policies	21-69
Redirection Policies	21-71
Policy Based Mirroring	21-72
ICMP Policy Example	21-72
802.1p and ToS/DSCP Marking and Mapping	21-73
Policy Based Routing	21-74
Chapter 22 Managing Policy Servers	22-1
In This Chapter	22-1
Policy Server Specifications	22-2
Policy Server Defaults	22-2
Policy Server Overview	22-3
Installing the LDAP Policy Server	22-3
Modifying Policy Servers	22-4
Modifying LDAP Policy Server Parameters	22-4
Disabling the Policy Server From Downloading Policies	22-4
Modifying the Port Number	22-5
Modifying the Policy Server Username and Password	22-5
Modifying the Searchbase	22-5
Configuring a Secure Socket Layer for a Policy Server	22-6
Loading Policies From an LDAP Server	22-6
Removing LDAP Policies From the Switch	22-6
Interaction With CLI Policies	22-7
Verifying the Policy Server Configuration	22-7

Chapter 23	Managing Authentication Servers	23-1
	In This Chapter	23-1
	Authentication Server Specifications	23-2
	Server Defaults	23-3
	RADIUS Authentication Servers	23-3
	TACACS+ Authentication Servers	23-3
	LDAP Authentication Servers	23-3
	Quick Steps For Configuring Authentication Servers	23-4
	Server Overview	23-5
	Backup Authentication Servers	23-5
	Authenticated Switch Access	23-5
	RADIUS Servers	23-7
	RADIUS Server Attributes	23-7
	Configuring the RADIUS Client	23-11
	TACACS+ Server	23-12
	TACACS+ Client Limitations	23-12
	Configuring the TACACS+ Client	23-13
	LDAP Servers	23-14
	Setting Up the LDAP Authentication Server	23-14
	LDAP Server Details	23-15
	Directory Server Schema for LDAP Authentication	23-20
	Configuring the LDAP Authentication Client	23-25
	Verifying the Authentication Server Configuration	23-27
Chapter 24	Configuring Port Mapping	24-1
	In This Chapter	24-1
	Port Mapping Specifications	24-2
	Port Mapping Defaults	24-2
	Quick Steps for Configuring Port Mapping	24-2
	Creating/Deleting a Port Mapping Session	24-3
	Creating a Port Mapping Session	24-3
	Deleting a Port Mapping Session	24-3
	Enabling/Disabling a Port Mapping Session	24-4
	Enabling a Port Mapping Session	24-4
	Disabling a Port Mapping Session	24-4
	Disabling the Flooding of Unknown Unicast Traffic	24-4
	Configuring a Port Mapping Direction	24-4
	Configuring Unidirectional Port Mapping	24-4
	Restoring Bidirectional Port Mapping	24-4
	Sample Port Mapping Configuration	24-5
	Example Port Mapping Overview	24-5
	Example Port Mapping Configuration Steps	24-6
	Verifying the Port Mapping Configuration	24-6

Chapter 25	Configuring Learned Port Security	25-1
	In This Chapter	25-1
	Learned Port Security Specifications	25-2
	Learned Port Security Defaults	25-2
	Sample Learned Port Security Configuration	25-3
	Learned Port Security Overview	25-5
	How LPS Authorizes Source MAC Addresses	25-6
	Dynamic Configuration of Authorized MAC Addresses	25-6
	Static Configuration of Authorized MAC Addresses	25-7
	Understanding the LPS Table	25-7
	Port and Link Aggregate Security Violation	25-8
	Configuring Learned Port Security	25-9
	Enabling/Disabling Learned Port Security	25-9
	Configuring a Source Learning Time Limit	25-10
	Configuring the Number of Bridged MAC Addresses Allowed	25-11
	Configuring the Number of Filtered MAC Addresses Allowed	25-11
	Configuring an Authorized MAC Address Range	25-12
	Selecting the Security Violation Mode	25-12
	Displaying Learned Port Security Information	25-13
Chapter 26	Diagnosing Switch Problems	26-1
	In This Chapter	26-1
	Port Mirroring Overview	26-3
	Port Mirroring Specifications	26-3
	Port Mirroring Defaults	26-3
	Quick Steps for Configuring Port Mirroring	26-4
	Port Monitoring Overview	26-5
	Port Monitoring Specifications	26-5
	Port Monitoring Defaults	26-5
	Quick Steps for Configuring Port Monitoring	26-6
	sFlow Overview	26-7
	sFlow Specifications	26-7
	sFlow Defaults	26-7
	Quick Steps for Configuring sFlow	26-8
	Remote Monitoring (RMON) Overview	26-10
	RMON Specifications	26-10
	RMON Probe Defaults	26-11
	Quick Steps for Enabling/Disabling RMON Probes	26-11
	Switch Health Overview	26-12
	Switch Health Specifications	26-12
	Switch Health Defaults	26-13
	Quick Steps for Configuring Switch Health	26-13
	Port Mirroring	26-14
	What Ports Can Be Mirrored?	26-14
	How Port Mirroring Works	26-14

What Happens to the Mirroring Port	26-15
Mirroring on Multiple Ports	26-15
Using Port Mirroring with External RMON Probes	26-15
Remote Port Mirroring	26-17
Creating a Mirroring Session	26-18
Unblocking Ports (Protection from Spanning Tree)	26-19
Enabling or Disabling Mirroring Status	26-19
Disabling a Mirroring Session (Disabling Mirroring Status)	26-19
Configuring Port Mirroring Direction	26-20
Enabling or Disabling a Port Mirroring Session (Shorthand)	26-20
Displaying Port Mirroring Status	26-21
Deleting A Mirroring Session	26-21
Configuring Remote Port Mirroring	26-22
Port Monitoring	26-23
Configuring a Port Monitoring Session	26-24
Enabling a Port Monitoring Session	26-24
Disabling a Port Monitoring Session	26-25
Deleting a Port Monitoring Session	26-25
Pausing a Port Monitoring Session	26-25
Configuring Port Monitoring Session Persistence	26-25
Configuring a Port Monitoring Data File	26-25
Configuring Port Monitoring Direction	26-26
Configuring capture-type	26-27
Displaying Port Monitoring Status and Data	26-27
sFlow	26-29
sFlow Manager	26-29
Receiver	26-29
Sampler	26-30
Poller	26-30
Configuring a sFlow Session	26-30
Configuring a Fixed Primary Address	26-31
Displaying a sFlow Receiver	26-31
Displaying a sFlow Sampler	26-32
Displaying a sFlow Poller	26-32
Displaying a sFlow Agent	26-33
Deleting a sFlow Session	26-33
Remote Monitoring (RMON)	26-34
Enabling or Disabling RMON Probes	26-36
Displaying RMON Tables	26-37
Monitoring Switch Health	26-41
Configuring Resource Thresholds	26-43
Displaying Health Threshold Limits	26-44
Configuring Sampling Intervals	26-45
Viewing Sampling Intervals	26-45
Viewing Health Statistics for the Switch	26-46
Viewing Health Statistics for a Specific Interface	26-47

Chapter 27	Configuring VLAN Stacking	27-1
	In This Chapter	27-1
	VLAN Stacking Specifications	27-2
	VLAN Stacking Defaults	27-2
	VLAN Stacking Overview	27-3
	How VLAN Stacking Works	27-5
	VLAN Stacking Services	27-6
	Interaction With Other Features	27-7
	Quick Steps for Configuring VLAN Stacking	27-8
	Configuring VLAN Stacking Services	27-10
	Configuring SVLANs	27-11
	Configuring a VLAN Stacking Service	27-12
	Configuring VLAN Stacking Network Ports	27-13
	Configuring a VLAN Stacking Service Access Point	27-14
	Configuring VLAN Stacking User Ports	27-15
	Configuring the Type of Customer Traffic to Tunnel	27-16
	Configuring a Service Access Point Profile	27-17
	Configuring a UNI Profile	27-19
	VLAN Stacking Application Example	27-20
	VLAN Stacking Configuration Example	27-21
	Verifying the VLAN Stacking Configuration	27-24
Chapter 28	Using Switch Logging	28-1
	In This Chapter	28-1
	Switch Logging Specifications	28-2
	Switch Logging Defaults	28-3
	Quick Steps for Configuring Switch Logging	28-4
	Switch Logging Overview	28-5
	Switch Logging Commands Overview	28-6
	Enabling Switch Logging	28-6
	Setting the Switch Logging Severity Level	28-6
	Specifying the Switch Logging Output Device	28-9
	Configuring the Switch Logging File Size	28-11
	Clearing the Switch Logging Files	28-11
	Displaying Switch Logging Records	28-12
	Alcatel-Lucent License Agreement	A-1
	ALCATEL-LUCENT SOFTWARE LICENSE AGREEMENT	A-1
	Third Party Licenses and Notices	A-4
	Index	Index-1



About This Guide

This *OmniSwitch AOS Release 7 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 OmniSwitch 10K switches.

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 *OmniSwitch 10K Getting Started 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 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 10K Getting Started Guide*
Release Notes

A hard-copy *OmniSwitch 10K Getting Started Guide* is included with your switch; 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 10K Hardware Users Guide*
OmniSwitch AOS Release 7 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 *OmniSwitch 10K Hardware Guide*. This guide provides 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.

This 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 7 Network Configuration Guide*
OmniSwitch AOS Release 7 Advanced Routing Configuration 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. This guide contains overview information, procedures, and examples on how standard networking technologies are configured on the OmniSwitch.

The guide includes configuration information for networks using advanced routing technologies (OSPF and BGP) and multicast routing protocols (DVMRP and PIM-SM).

Anytime

The *OmniSwitch 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 OmniSwitch 10K user manuals:

- *OmniSwitch 10K Getting Started Guide*

Describes the hardware and software procedures for getting an OmniSwitch up and running. Also provides information on fundamental aspects of OmniSwitch software architecture.

- *OmniSwitch 10K Hardware Users Guide*

Complete technical specifications and procedures for all OmniSwitch 10K chassis, power supplies, fans, and Network Interface (NI) modules.

- *OmniSwitch 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 7 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 7 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 7 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 Transceivers Guide*

Includes SFP and XFP transceiver specifications and product compatibility information.

- *Technical Tips, Field Notices*

Includes information published by Alcatel'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 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 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'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's technical support, open a new case or access helpful release notes, technical bulletins, and manuals.

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

Web: service.esd.alcatel-lucent.com

Phone: 1-800-995-2696

Email: esd.support@alcatel-lucent.com

1 Configuring Ethernet Ports

The Ethernet software is responsible for a variety of functions that support Ethernet, Gigabit Ethernet, and 10 Gigabit 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:

- [“Ethernet Ports Overview” on page 1-3](#)
- [“Configuring Ethernet Port Parameters” on page 1-3](#)

For information about CLI commands that can be used to view Ethernet port parameters, see the *OmniSwitch CLI Reference Guide*.

Ethernet Specifications

IEEE Standards Supported	802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) 802.3u (100BaseTX) 802.3ab (1000BaseT) 802.3z (1000Base-X) 802.3ae (10GBase-X)
Platforms Supported	OmniSwitch 10K
Ports Supported	Ethernet (10 Mbps) Fast Ethernet (100 Mbps) Gigabit Ethernet (1 Gb/1000 Mbps) 10 Gigabit Ethernet (10 Gb/10000 Mbps)
Auto Negotiation	Supported
Port Mirroring / Monitoring	Supported
802.1Q Hardware Tagging	Supported
Jumbo Frame Configuration	Supported on Gigabit Ethernet and 10 Gigabit Ethernet ports
Maximum Frame Size	1553 bytes (10/100 Mbps) 9216 bytes (1/10 Gbps)

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 (10 Ethernet) 49 Mbps (100 Fast Ethernet) 496 Mbps (1 Gigabit Ethernet) 997 Mbps (10 Gigabit 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

Ethernet Ports Overview

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

Valid Port Settings

This table below lists valid speed, duplex, and autonegotiation settings for the different OS10K port types.

Port Type	User-Specified Port Speed (Mbps) Supported	User-Specified Duplex Supported	Auto Negotiation Supported?
10/100/1000 Copper	auto/10/100/1000	auto/full/half	Yes
Transceivers	Dependent	Dependent	Dependent

See the *OS10K Hardware Users Guide* for more information about the OS10K hardware and port numbering for specific models.

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

Resetting Statistics Counters

The **clear interfaces l2-statistics** 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 following section describes how to apply a flood limit value to broadcast, unicast flooded, or multi-cast traffic for a slot, port, or a range of ports. The **interfaces flood-limit** command can be used to set limits based on pps, mbps, or a percentage of the port's bandwidth.

For example, to set the broadcast limit on port 2/1 to 100 mbps enter the following:

```
-> interfaces 2/1 flood-limit bcast rate mbps 100
```

For example, to set the flood limit on ports 2/2-5 to 500 pps enter the following:

```
-> interfaces 2/2-5 flood-limit uucast rate pps 500
```

For example, to set the multicast limit on slot 3 to 50% of the ports capacity enter the following:

```
-> interfaces 3 flood-limit mcast rate cap% 50
```

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. The OmniSwitch currently does not support the transmitting of PAUSE frames.

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
```


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 operates at Layer 2 in conjunction with 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 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 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](#) to configure the probe-message advertisement timer.
- [“Configuring the Echo-Wait-Timer” on page 2-7](#) to configure the echo-based detection timer.
- [“Clearing UDLD Statistics” on page 2-8.](#)
- [“Verifying the UDLD Configuration” on page 2-8.](#)
- [“Verifying the UDLD Configuration” on page 2-8.](#)

UDLD Specifications

Platforms Supported	OmniSwitch 10K
Maximum number of UDLD ports per system	Up to maximum physical ports per system

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 the creation of 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 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 uddl statistics port** command. For example, to clear the statistics for port 4 on slot 1, enter:

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

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

```
-> clear uddl statistics
```

Verifying the UDLD Configuration

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

show uddl configuration	Displays the global status of UDLD configuration.
show uddl configuration port	Displays the configuration information for all UDLD ports or for a particular UDLD port on the switch.
show uddl statistics port	Displays the UDLD statistics for a specific port.
show uddl neighbor port	Displays the UDLD neighbor ports.
show uddl 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 CLI Reference Guide*. An example of the output for the **show uddl configuration port** and **show uddl 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 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 Specifications

The functionality described in this chapter is supported on the OmniSwitch unless otherwise stated in the following Specifications table or specifically noted within any section of this chapter.

Platforms Supported	OmniSwitch 10K
RFCs supported	2674—Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions
Maximum number of learned MAC addresses when centralized MAC source learning mode is enabled	32K Module / 32K Chassis
Maximum number of learned MAC addresses when distributed MAC source learning mode is enabled.	32K Module / 256K Chassis

Source Learning Defaults

Parameter Description	Command	Default
Static MAC address operating mode	mac-learning vlan 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 vlan 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 vlan 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 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 6, “Configuring Static Link Aggregation”](#) and [Chapter 7, “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 vlan 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 vlan 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 vlan 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 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 vlan multi-cast 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 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).
- 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 distributed MAC source learning 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 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 Alcatel-Lucent switching 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 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 Router Interfaces” on page 4-10.](#)
- [“Bridging VLANs Across Multiple Switches” on page 4-11.](#)
- [“Verifying the VLAN Configuration” on page 4-13.](#)

For information about Spanning Tree, see [Chapter 5, “Configuring Spanning Tree Parameters.”](#)

For information about routing, see [Chapter 11, “Configuring IP.”](#)

VLAN Specifications

Note. The maximum limit values provided in the following Specifications table are subject to available system resources.

Platforms Supported	OmniSwitch 10K
RFCs Supported	2674 - Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions
IEEE Standards Supported	802.1Q - Virtual Bridged Local Area Networks 802.1D - Media Access Control Bridges
Maximum VLANs per switch	4094
Maximum Tagged VLANs per Port	4094
Maximum Untagged VLANs per Port	One untagged VLAN (default VLAN) per port.
Maximum VLAN Port Associations (VPA) per switch	20000
Maximum Spanning Tree VLANs per switch	128 (1x1 mode)

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 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 router interface using the following command to assign an IP host address of 21.0.0.10 to VLAN 100 that enables routing 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 Alcatel-Lucent switches 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 related to the assigned 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 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 Router 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 4-7](#).
- 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 6, “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.

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 6, "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.

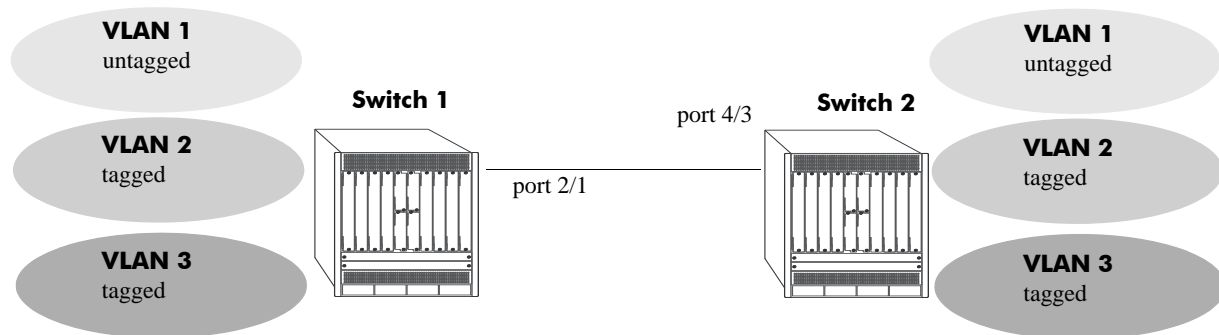
The OmniSwitch implements the IEEE 802.1Q standard for sending frames through the network tagged with VLAN identification. This section details procedures for configuring and monitoring 802.1Q tagging on a single switch port or link aggregate group.

"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:



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. 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 6, “Configuring Static Link Aggregation”](#) or [Chapter 7, “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 set 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/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 5, “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/disable source learning on a VLAN, use the `mac-learning vlan static mac-address` command. For example, the following command disabled source learning on VLAN 10:

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

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

Configuring VLAN Router 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, IP - 21.0.0.10.

Alcatel-Lucent switches 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. Up to eight IP interfaces can be configured for each VLAN.

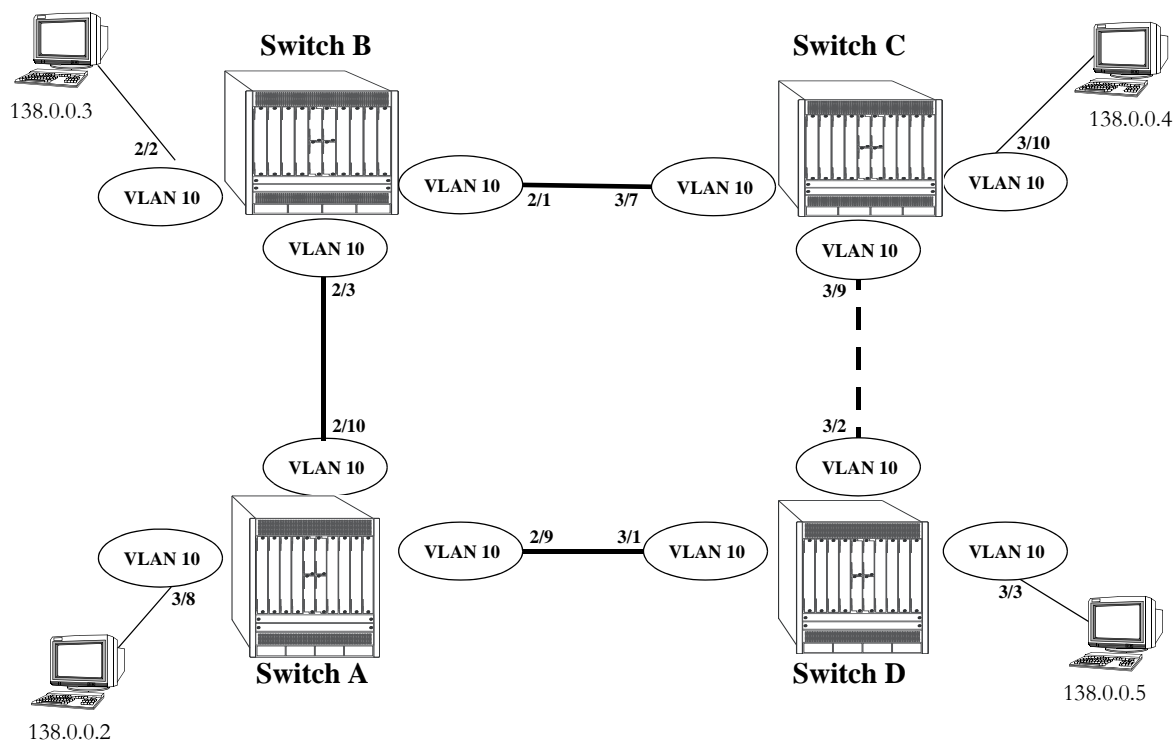
If a VLAN does not have a router interface, the ports associated with that VLAN are in essence firewalled from other VLANs. For information about how to configure router interfaces, see [Chapter 11, “Configuring IP”](#).

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:



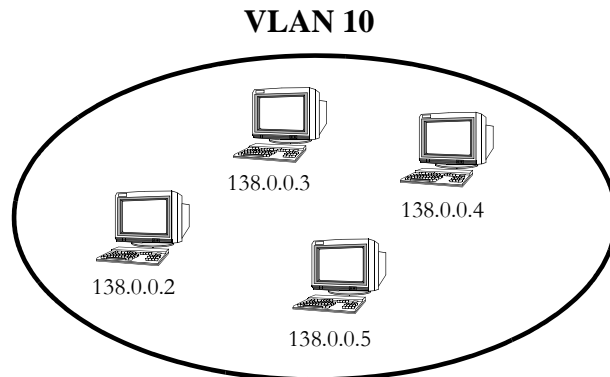
VLAN Bridging Domain: Physical Configuration

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 5, “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.



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

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 show vlan members untagged command. The VLAN is now the port's configured default VLAN.
qtagged	The port was statically assigned to the VLAN using the show 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 26, "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 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 CLI Reference Guide*.

5 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 Alcatel-Lucent 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.

Alcatel-Lucent's distributed 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.

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 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Selecting the Spanning Tree operating mode (flat or per-VLAN) on [page 5-20](#).
- Configuring Spanning Tree bridge parameters on [page 5-25](#).
- Configuring Spanning Tree port parameters on [page 5-33](#).
- Configuring an example Spanning Tree topology on [page 5-44](#).

Spanning Tree Specifications

Platforms Supported	OmniSwitch 10K
IEEE Standards supported	802.1d—Media Access Control (MAC) Bridges 802.1s—Multiple Spanning Trees 802.1w—Rapid Spanning Tree Protocol
Spanning Tree operating modes supported	Flat mode—one spanning tree instance per switch Per-VLAN mode—one spanning tree instance per VLAN
Spanning Tree port eligibility	Fixed ports 802.1Q tagged ports Link aggregate of ports
Maximum VLAN Spanning Tree instances per switch.	128 (per-VLAN mode)
Maximum flat mode Multiple Spanning Tree Instances (MSTI) per switch	16 MSTI, in addition to the Common and Internal Spanning Tree instance (also referred to as MSTI 0).

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 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 vlan mode	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

Alcatel-Lucent switches support 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 an Alcatel-Lucent 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 5-12](#) 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 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 5-37 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 5-27 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 5-36 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:

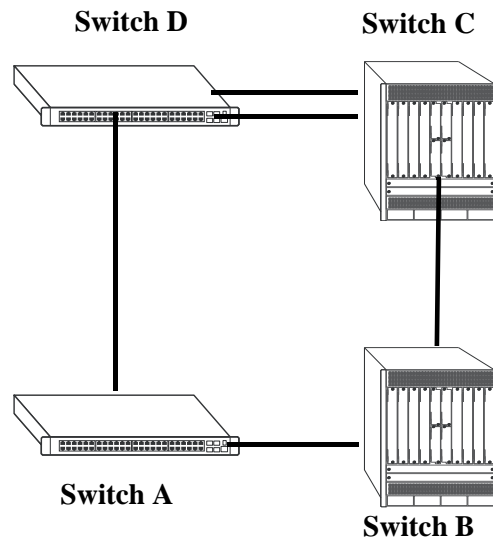
- 1** The lowest root bridge ID (lowest priority value, then lowest MAC address).
- 2** The best root path cost.
- 3** If root path costs are equal, the bridge ID of the bridge sending the BPDU.
- 4** If the previous three values tie, then the port ID (lowest priority value, then lowest port number).

When a topology change occurs, such as when a link goes down or a switch is added to the network, the affected bridge sends 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. To restrict TCN propagation on a port, see [“Configuring STP Port Parameters” on page 5-33](#).

Topology Examples

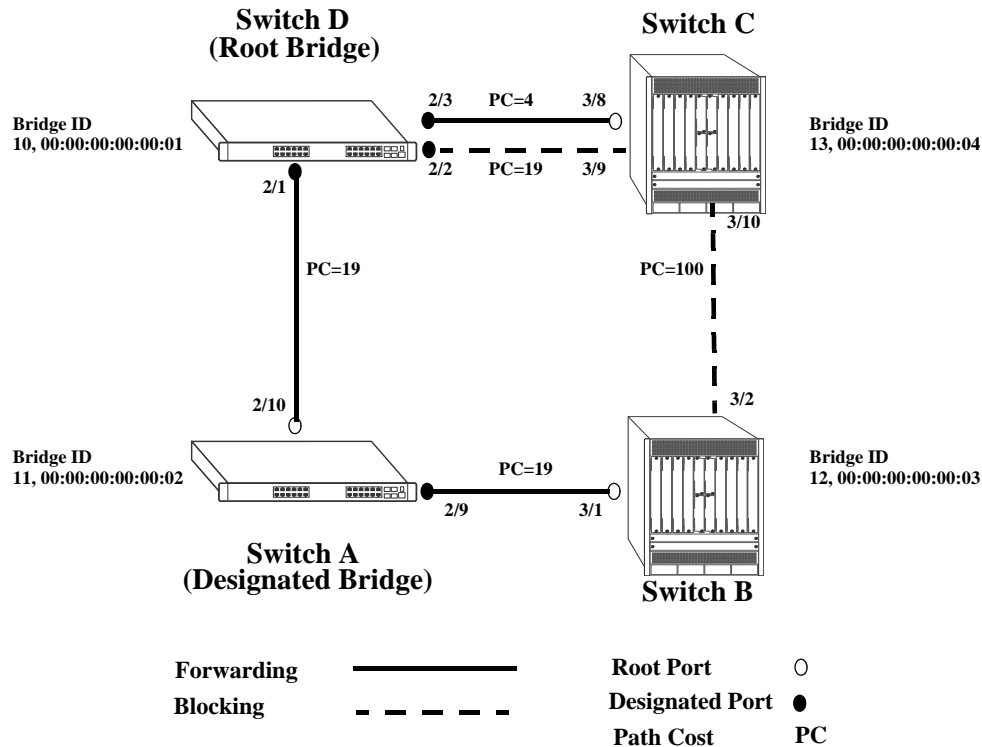
The following diagram shows an example of a physical network topology that incorporates data path redundancy to ensure fault tolerance. These redundant paths, however, create loops in the network configuration. If a device connected to Switch A sends broadcast packets, Switch A floods the packets out 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*.



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 transitions 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:



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 5-12.](#)
- [“Comparing MSTP with STP and RSTP” on page 5-15.](#)
- [“What is a Multiple Spanning Tree Instance \(MSTI\)” on page 5-15.](#)
- [“What is a Multiple Spanning Tree Region” on page 5-16.](#)
- [“What is the Internal Spanning Tree \(IST\) Instance” on page 5-17.](#)
- [“What is the Common and Internal Spanning Tree Instance” on page 5-17.](#)
- [“MST Configuration Overview” on page 5-17.](#)

How MSTP Works

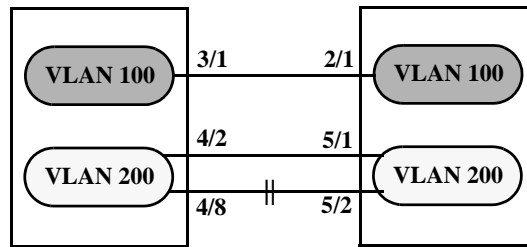
MSTP, as defined in the IEEE 802.1Q 2005 standard, is an enhancement to the IEEE 802.1Q Common Spanning Tree (CST). The CST is a single spanning tree that uses 802.1D (STP) or 802.1w (RSTP) to provide a loop-free network topology.

The Alcatel-Lucent flat spanning tree mode applies a single CST instance on a per switch basis. The per-VLAN mode is an Alcatel-Lucent 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 Alcatel-Lucent MSTP flat mode switches, 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.



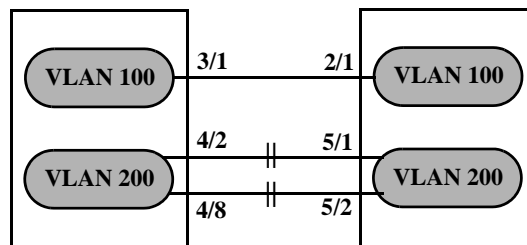
Per-VLAN Mode STP/RSTP

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.

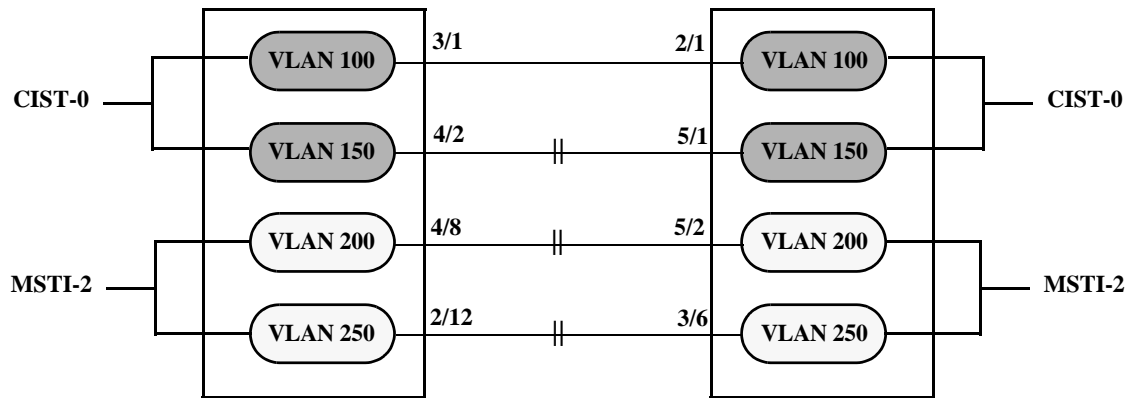
- The connections between 4/8 and 5/2 and 4/2 and 5/1 are seen as redundant because they are both controlled by the VLAN 200 Spanning Tree instance and connect to the same switches. The VLAN 200 Spanning Tree instance determines which connection provides the best data path and transitions the other connection to a blocking state.



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.



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 5-49](#) 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 5-15](#) 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 5-17](#) 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 5-16](#) 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. Alcatel-Lucent switches support 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 5-17](#) for more information.

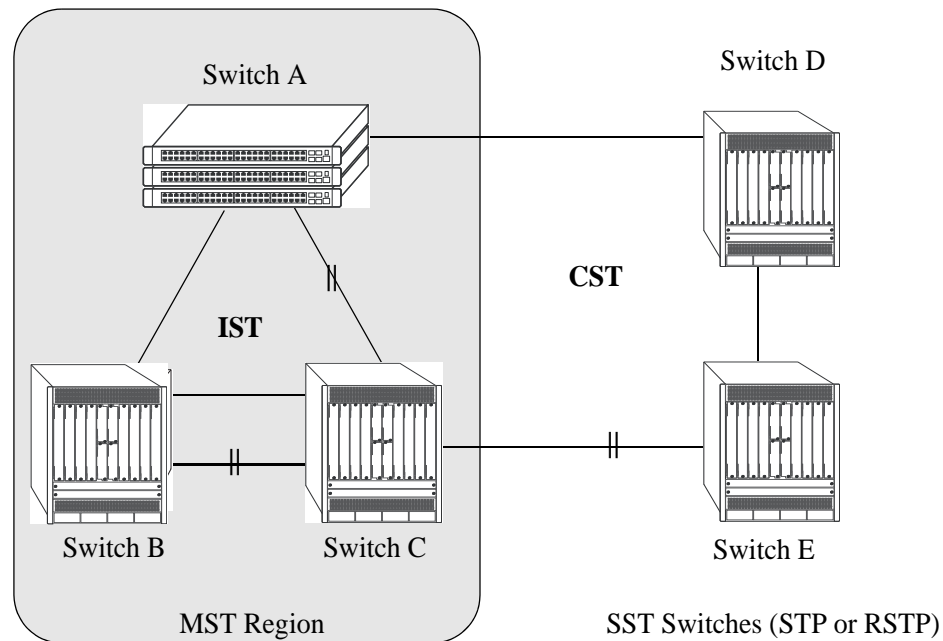
What is a Multiple Spanning Tree Region

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 5-17](#) for more information.



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 5-47](#) 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. 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 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 5-25](#) 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 5-20](#) 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 5-12](#) 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 5-16](#) 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 5-15 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 5-47 and [“Sample MSTI Configuration”](#) on page 5-49.

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 if necessary. 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 5-12 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 5-25](#) 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 Alcatel-Lucent proprietary implementation that applies one Spanning Tree instance to each VLAN. For example, if five VLANs exist on the switch, then there 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 if necessary. 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 5-12](#) 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

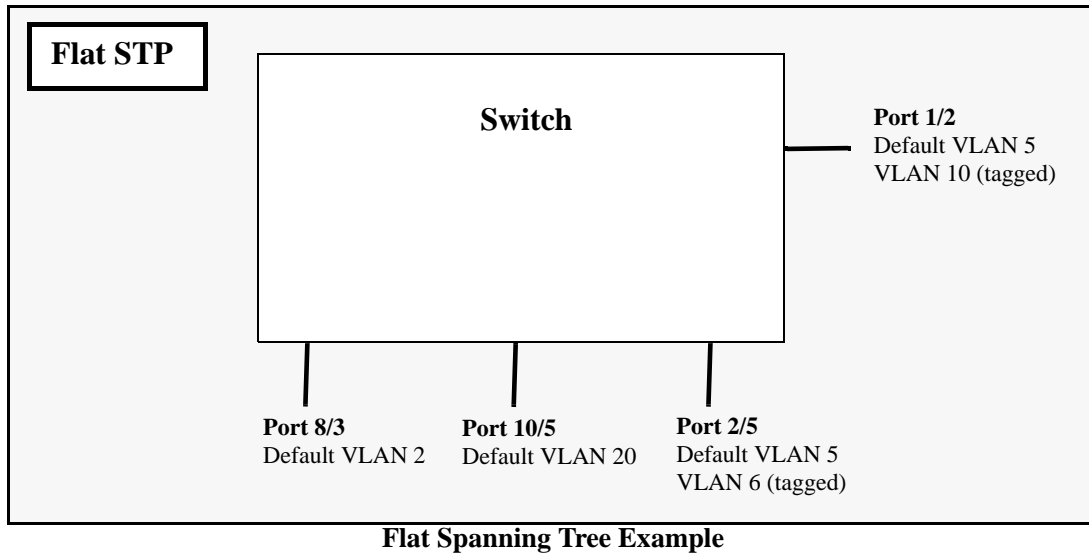
Before selecting the flat Spanning Tree mode, consider the following:

- 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 [Chapter 5, “Configuring Spanning Tree Parameters.”](#)



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

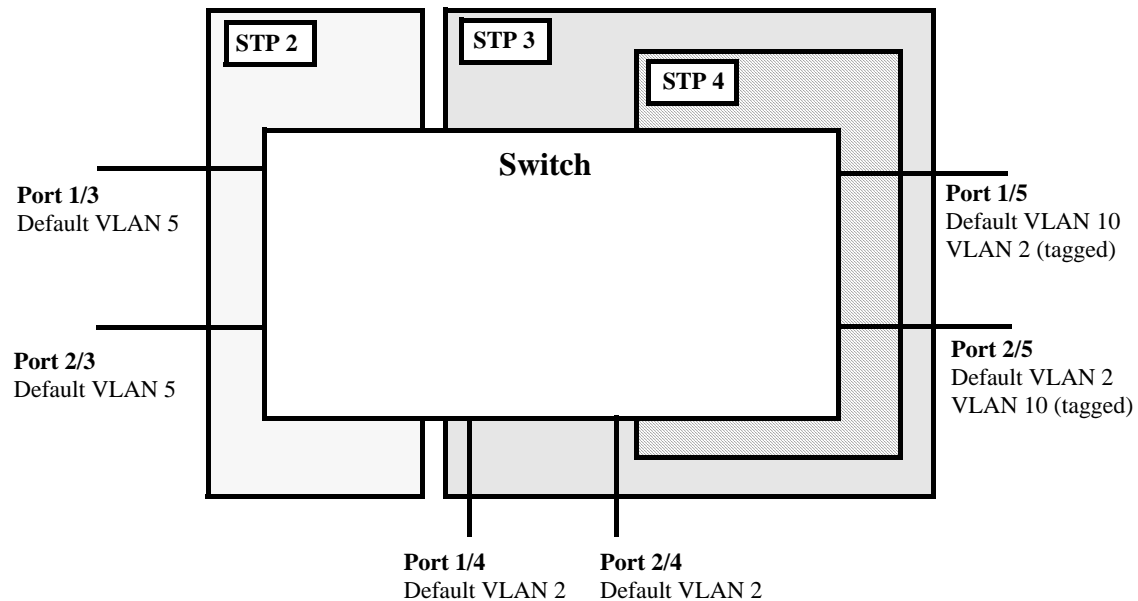
Before selecting the Per-VLAN Spanning Tree operating mode, consider the following:

- 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.



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 current Alcatel-Lucent 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 Alcatel-Lucent 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 Alcatel-Lucent 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 Alcatel-Lucent *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 [“Configuring the Bridge Priority” on page 5-27](#).
- 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 [“Configuring the Bridge Priority” on page 5-27](#).
- 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 [“Configuring the Path Cost Mode” on page 5-31](#).
- 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 [“Path Cost for Link Aggregate Ports” on page 5-38](#).

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 Alcatel-Lucent 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 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.

Commands	Used for ...
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 the 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:

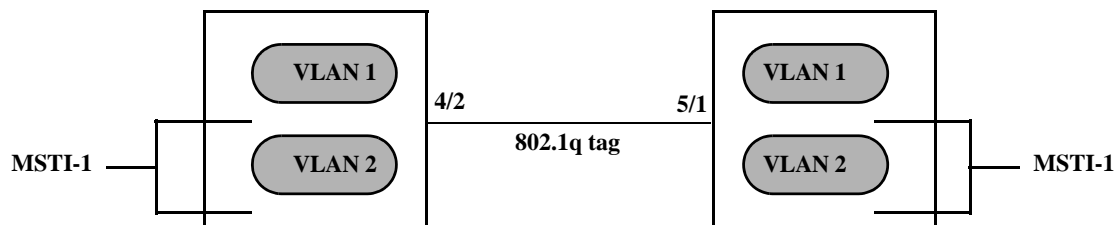
```
-> spantree path-cost-mode 32bit
```

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

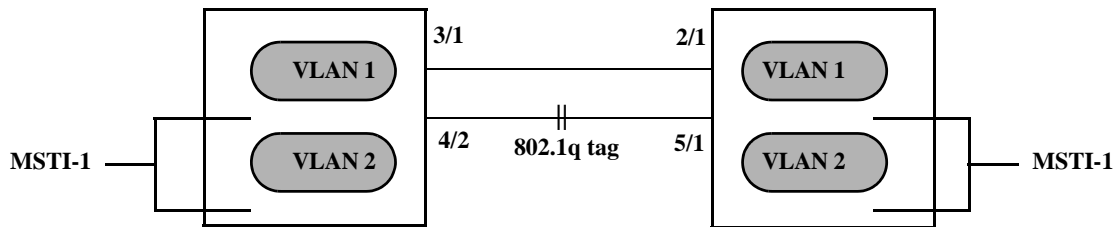
In a Multiple Spanning Tree (MST) configuration, it is possible for a port that belongs to a VLAN that is not a member of an instance to become the root port for that instance. This can cause a topology change that could lead to a loss of connectivity between VLANs/switches. Enabling Automatic VLAN Containment (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.



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:



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 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 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 vlan mode	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 6, “Configuring Static Link Aggregation,”](#) and [Chapter 7, “Configuring Dynamic Link Aggregation.”](#)

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 6, “Configuring Static Link Aggregation,”](#) and [Chapter 7, “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 5-31](#) for more information.

If a 32-bit path cost value is in use and the *path_cost* is set to zero, the following IEEE 802.1Q 2005 recommended default path cost values based on link speed are used:

Link Speed	IEEE 802.1D Recommended Value
10 MB	2,000,000
100 MB	200,000
1 GB	20,000
10 Gbps	2,000

If 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	IEEE 802.1D Recommended Value
4 Mbps	250
10 Mbps	100
16 Mbps	62
100 Mbps	19
1 Gbps	4
10 Gbps	2

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.

If a 32-bit path cost value is in use and the *path_cost* for a link aggregate is set to zero, the following default values based on link speed and link aggregate size are used:

Link Speed	Aggregate Size (number of links)	Default Path Cost Value
10 MB	2	1,200,000
	4	800,000
	8	600,000
100 MB	2	120,000
	4	80,000
	8	60,000
1 GB	2	12,000
	4	8,000
	8	6,000
10 GB	2	1,200
	4	800
	8	600

If a 16-bit path cost value is in use and the *path_cost* for a link aggregate is set to zero, the following default values based on link speed and link aggregate size are used. Note that for Gigabit ports the aggregate size is not applicable in this case:

Link Speed	Aggregate Size (number of links)	Default Path Cost Value
10 Mbps	2	60
	4	40

Link Speed	Aggregate Size (number of links)	Default Path Cost Value
	8	30
100 Mbps	2	12
	4	9
	8	7
1 Gbps	N/A	3
10 Gbps	N/A	1

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 6, “Configuring Static Link Aggregation,”](#) and [Chapter 7, “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 vlan mode** 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 6, “Configuring Static Link Aggregation,”](#) and [Chapter 7, “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 5-42](#) 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 6, “Configuring Static Link Aggregation,”](#) and [Chapter 7, “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
```

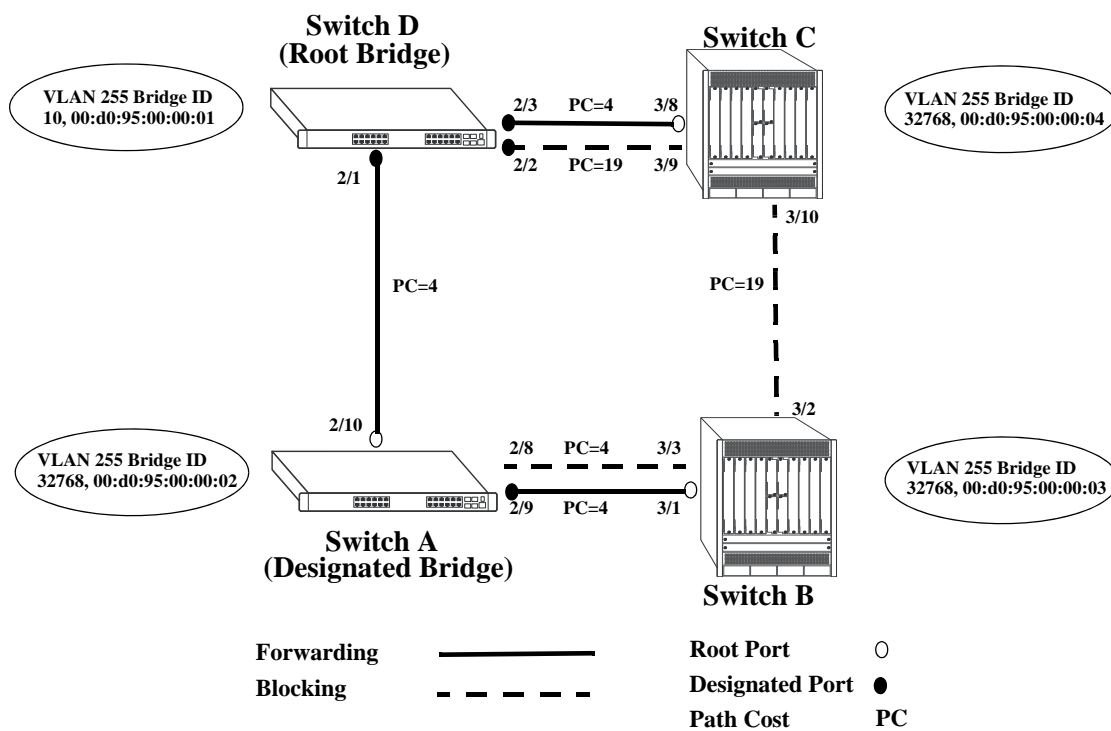
Sample Spanning Tree Configuration

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 5-12 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:



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 5-44](#).

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 5-44](#) 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:

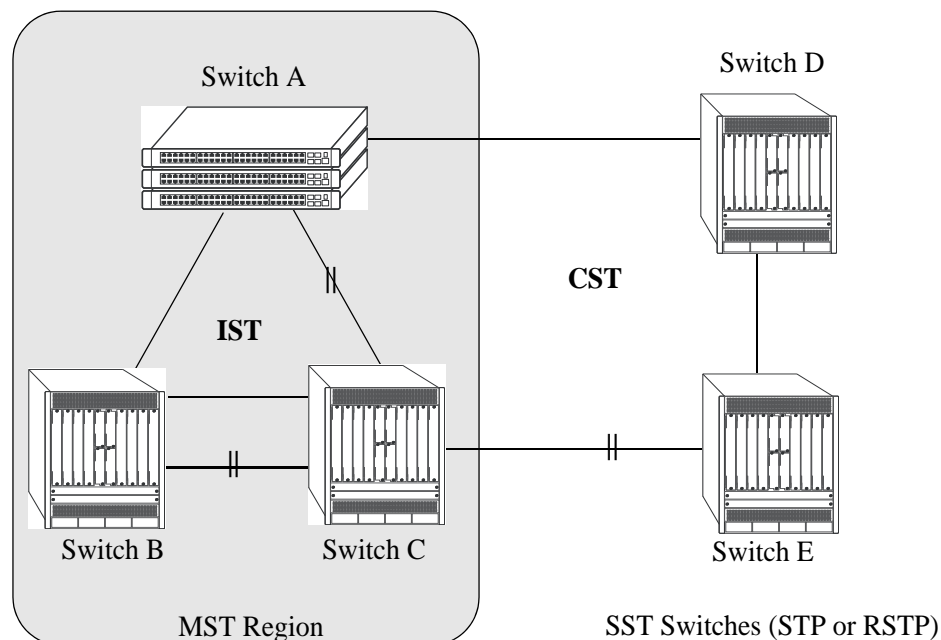
```
-> spantree vlan 255 protocol rstp
```


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:



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 **Alcatel-Lucent 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 "Alcatel 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 5-49 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      = Alcatel Marketing,
Revision Level          = 2000,
Configuration Digest    = 0x922fb3f 31752d68 67fe1155 d0ce8380,
Revision Max hops       = 3,
Cist Instance Number    = 0
```

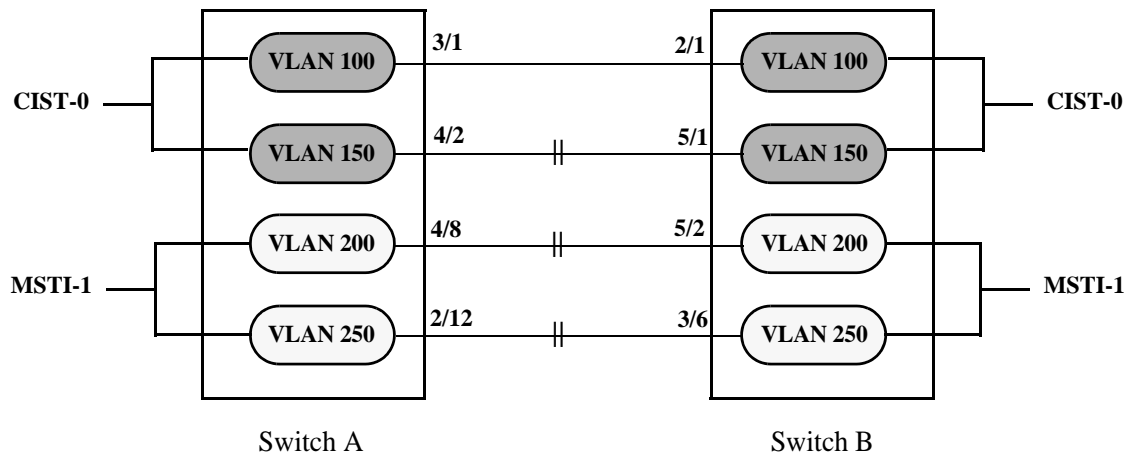
All switches configured with the exact same values as shown in the above example are considered members of the Alcatel-Lucent Marketing MST region.

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:



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 5-25 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
```

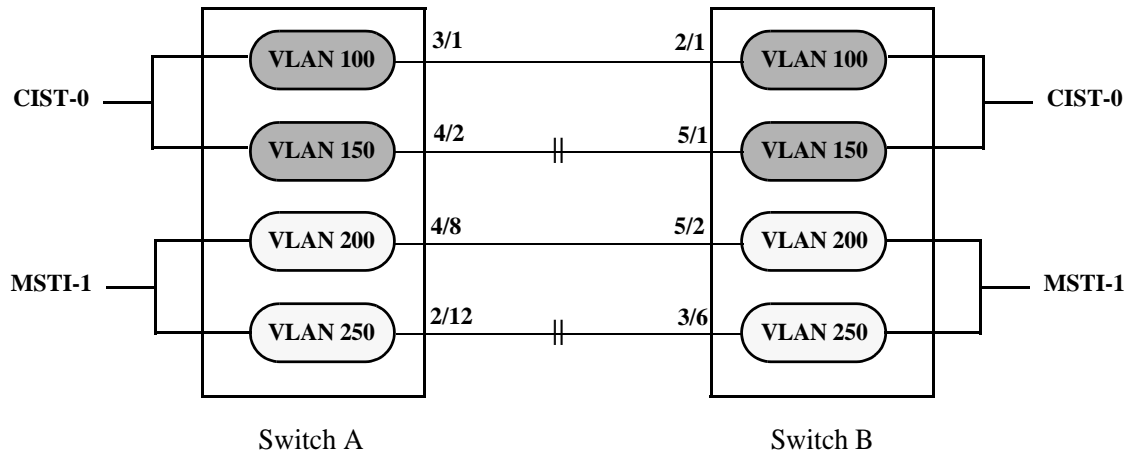
All VLANs are associated with the CIST instance. As a result, VLANs 100 and 150 do not require any configuration to map them to the CIST instance.

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 5-49](#). 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 5-51](#).



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 5-12](#) 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 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 5-45.

6 Configuring Static Link Aggregation

Alcatel-Lucent's 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 “[Static Link Aggregation Specifications](#)” table that consist of 2, 4, or 8 Ethernet links.
- **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 from 100-Mbps Ethernet backbones to Gigabit Ethernet 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 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](#)”.

Note. You can also configure and monitor static link aggregation with WebView, Alcatel-Lucent's embedded web-based device management application. WebView is an interactive and easy-to-use GUI that can be launched from OmniVista or a web browser. Please refer to the WebView online documentation for more information on configuring and monitoring static link aggregation with WebView.

Static Link Aggregation Specifications

The table below lists specifications for static groups.

Platforms Supported	OmniSwitch 10000
Maximum number of link aggregation groups	128
Maximum number of links per group supported	8
Maximum number of linkagg ports per system	256

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 6-11](#) 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 2, 4, or 8 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 OmniSwitch AOS software treats these virtual links just like physical links. (See [“Relationship to Other Features”](#) on page 6-6 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.

Alcatel-Lucent’s 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 7, “Configuring Dynamic Link Aggregation.”](#)

Static Link Aggregation Operation

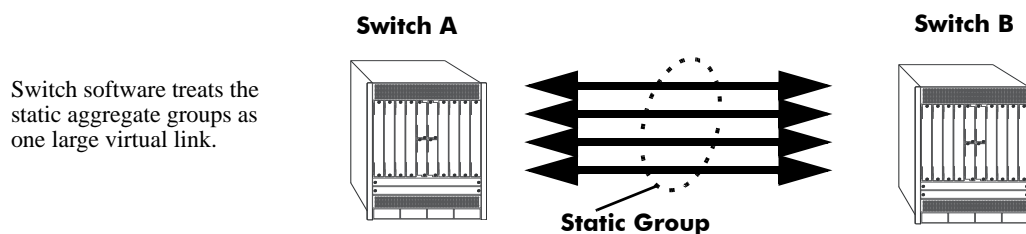
Static link aggregate groups are virtual links between two nodes consisting of 2, 4, or 8 fixed physical links.

Static aggregate groups can be created between each of the following OmniSwitch products:

- two OmniSwitch 10000 switches.
- an OmniSwitch 10000 switch and an early-generation Alcatel-Lucent switch.

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 OS9-GNI-C24 on Switch A to two ports on one OS9-GNI-C24 and two ports on another OS9-GNI-C24 on Switch B. The network administrator has created a separate VLAN for this group so users can use this high speed link.



Example of a Static Link Aggregate Group Network

See [“Configuring Static Link Aggregation Groups”](#) on page 6-6 for information on using Command Line Interface (CLI) commands to configure static aggregate groups and see [“Displaying Static Link Aggrega-](#)

tion Configuration and Statistics” on page 6-11 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 6, “Configuring Static Link Aggregation.”](#)

Note. See [“Application Example” on page 6-10](#) 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 6-6](#) for more information.

Note. See [“Quick Steps for Configuring Static Link Aggregation” on page 6-3](#) for a brief tutorial on configuring these mandatory parameters.

Alcatel-Lucent’s 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 6-2](#). If you need to modify any of these parameters, please see [“Modifying Static Aggregation Group Parameters” on page 6-9](#) for more information.

Note. See the “Link Aggregation Commands” chapter in the *OmniSwitch 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 6-7](#).
- 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 6-8](#).

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 6-9.

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, which can be 2, 4, or 8.

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 must 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 6-9)
- Static aggregate group administrative state (see “[Modifying the Static Aggregate Group Administrative State](#)” on page 6-9)

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

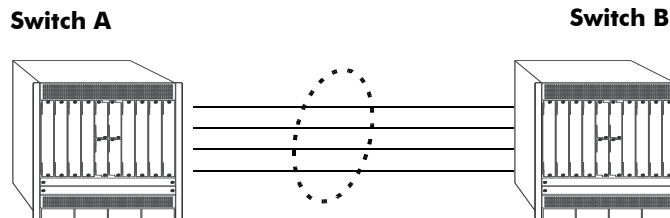
To disable a static aggregate group by entering [linkagg static agg](#) followed by the number of the group and **admin-state disable**. For example, to disable static aggregate group 1, enter:

```
-> 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.



Static Aggregate Group 1
VLAN 8 with 802.1Q tagging has been configured to use this group.

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
```

A screen similar to the following would be displayed:

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
```

A screen similar to the following would be displayed:

```
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
```

Note. See the “Link Aggregation Commands” chapter in the *OmniSwitch CLI Reference Guide* for complete documentation of **show** commands for link aggregation.

7 Configuring Dynamic Link Aggregation

Alcatel-Lucent's 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 up to a maximum number of link aggregation groups as mentioned in “[Dynamic Link Aggregation Specifications](#)” on page 7-2, that consist of 2, 4, or 8 Ethernet links
- **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 from 100-Mbps Ethernet backbones to Gigabit Ethernet 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 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Configuring dynamic link aggregation groups on “[Configuring Dynamic Link Aggregate Groups](#)” on page 7-8.
- 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 7-10.
- Modifying dynamic link aggregation parameters on “[Modifying Dynamic Link Aggregate Group Parameters](#)” on page 7-12.

Note. You can also configure and monitor dynamic link aggregation with WebView, Alcatel-Lucent's embedded Web-based device management application. WebView is an interactive and easy-to-use GUI that can be launched from OmniVista or a Web browser. Please refer to the WebView online documentation for more information on configuring and monitoring dynamic link aggregation with WebView.

Dynamic Link Aggregation Specifications

The table below lists specifications for dynamic aggregation groups and ports:

Platforms Supported	OmniSwitch 10K
IEEE Specifications Supported	802.3ad — Aggregation of Multiple Link Segments
Maximum number of link aggregation groups	128
Maximum number of ports per link aggregate	8
Number of ports supported per link aggregate group when 128 linkagg groups are configured	2
Maximum number of linkagg ports per system	256

Dynamic Link Aggregation Default Values

The table below lists default values for dynamic aggregate groups.

Parameter Description	Command	Default Value/Comments
Group Administrative State	<code>linkagg lacp agg admin-state</code>	enabled
Group Name	<code>linkagg lacp agg name</code>	No name configured
Group Actor Administrative Key	<code>linkagg lacp agg actor admin-key</code>	0
Group Actor System Priority	<code>linkagg lacp agg actor system-priority</code>	0
Group Actor System ID	<code>linkagg lacp agg actor system-id</code>	00:00:00:00:00:00
Group Partner System ID	<code>linkagg lacp agg partner system-id</code>	00:00:00:00:00:00
Group Partner System Priority	<code>linkagg lacp agg partner system-priority</code>	0
Group Partner Administrative Key	<code>linkagg lacp agg partner admin-key</code>	0
Actor Port Administrative State	<code>linkagg lacp agg admin-state</code>	active timeout aggregate
Actor Port System ID	<code>linkagg lacp agg actor system-id</code>	00:00:00:00:00:00
Partner Port System Administrative State	<code>linkagg lacp agg partner admin-state</code>	active timeout aggregate
Partner Port Admin System ID	<code>linkagg lacp port partner admin system-priority</code>	00:00:00:00:00:00
Partner Port Administrative Key	<code>linkagg lacp agg partner admin-key</code>	0
Partner Port Admin System Priority	<code>linkagg lacp agg partner system-priority</code>	0
Actor Port Priority	<code>linkagg lacp port actor port priority</code>	0
Partner Port Administrative Port	<code>linkagg lacp port partner admin-port</code>	0
Partner Port Priority	<code>linkagg lacp port partner admin port-priority</code>	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
```

When multi-chassis link aggregation feature is activated on the switch, the **show linkagg agg** command displays the output as MC-Dynamic aggregate.

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 7-30](#) for more information on **show** commands.

An example of what these commands look like entered sequentially on the command line on the actor switch:

```
-> 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 command line on the partner switch:

```
-> 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 2, 4, or 8 physical connections into large virtual connections known as link aggregation *groups*. You can configure up to 128 link aggregation groups for the switch and up to 128 groups for a chassis-based switch. Each group can consist of 2, 4, or 8 Ethernet links.

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 7-8 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.

Alcatel-Lucent’s link aggregation software allows you to configure the following two different types of link aggregation groups:

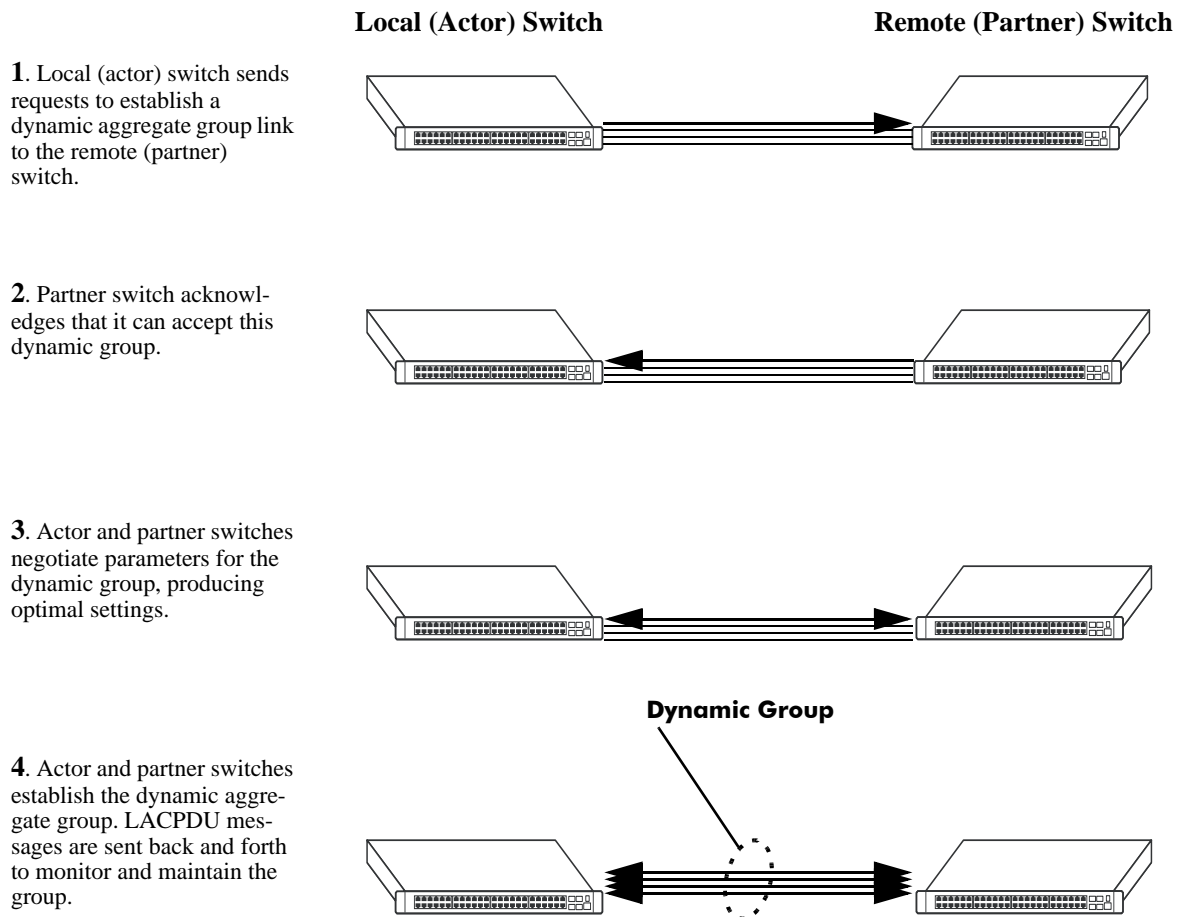
- Static link aggregate groups
- Dynamic link aggregate groups

This chapter describes dynamic link aggregation. For information on static link aggregation, please refer to [Chapter 6, “Configuring Static Link Aggregation.”](#)

Dynamic Link Aggregation Operation

Dynamic aggregate groups are virtual links between two nodes consisting of 2, 4, or 8 10-Mbps, 100-Mbps, or 1-or 10-Gbps fixed 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.



Example of a Dynamic Aggregate Group Network

Dynamic aggregate groups can be created between each of the following OmniSwitch products:

- two OmniSwitch 10K switches.
- an OmniSwitch 10K switch and an early-generation Alcatel-Lucent switch.
- an OmniSwitch 10K switch and switch of another vendor, if that vendor supports IEEE 802.3ad LACP.

See [“Configuring Dynamic Link Aggregate Groups” on page 7-8](#) 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 7-30](#) 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 5, “Configuring Spanning Tree Parameters.”](#)

Note. See [“Application Examples” on page 7-27](#) for tutorials on using link aggregation with other features.

Configuring Dynamic Link Aggregate Groups

This section describes how to use Alcatel-Lucent’s Command Line Interface (CLI) commands to create, modify, and delete dynamic aggregate groups. See [“Configuring Mandatory Dynamic Link Aggregate Parameters” on page 7-9](#) for more information.

Note. See [“Quick Steps for Configuring Dynamic Link Aggregation” on page 7-4](#) for a brief tutorial on configuring these mandatory parameters.

Alcatel-Lucent’s 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 7-3](#). For most configurations, using only the steps described in [“Creating and Deleting a Dynamic Aggregate Group” on page 7-9](#) 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 7-3](#), please see [“Modifying Dynamic Link Aggregate Group Parameters” on page 7-12](#) for more information.

Note. See the “Link Aggregation Commands” chapter in the *OmniSwitch 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 7-9.

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 7-10.

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 7-12. 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, which can be 2, 4, or 8. For example, to configure the dynamic aggregate group 2 consisting of eight links enter:

```
-> linkagg lacp agg 2 size 8
```

You can create up to 32 link aggregation (both static and dynamic) groups for a standalone switch and up to 128 groups for 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, Alcatel-Lucent recommends assigning the actor admin key when you create the dynamic aggregate group 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 7-13](#).

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 7-17](#) and [“Modifying the Partner Port System Administrative State” on page 7-21](#), 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 7-3](#) lists default group and port settings for Alcatel-Lucent Dynamic Link Aggregation software. 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 7-12](#)
- Dynamic aggregate actor ports on [page 7-17](#)
- Dynamic aggregate partner ports on [page 7-21](#)

Note. You *must* create a dynamic aggregate group before you can modify group or port parameters. See [“Configuring Dynamic Link Aggregate Groups” on page 7-8](#) 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 7-13](#))
- Group administrative state (see [“Modifying the Dynamic Aggregate Group Administrative State” on page 7-13](#))
- Group local (actor) switch actor administrative key (see [“Configuring and Deleting the Dynamic Aggregate Group Actor Administrative Key” on page 7-14](#))
- Group local (actor) switch system priority (see [“Modifying the Dynamic Aggregate Group Actor System Priority” on page 7-14](#))
- Group local (actor) switch system ID (see [“Modifying the Dynamic Aggregate Group Actor System ID” on page 7-15](#))
- Group remote (partner) administrative key (see [“Modifying the Dynamic Aggregate Group Partner Administrative Key” on page 7-15](#))
- Group remote (partner) system priority (see [“Modifying the Dynamic Aggregate Group Partner System Priority” on page 7-16](#))
- Group remote (partner) switch system ID (see [“Modifying the Dynamic Aggregate Group Partner System ID” on page 7-16](#))

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 7-17](#))
- Actor port system ID (see “[Modifying the Actor Port System ID](#)” on [page 7-19](#))
- Actor port system priority (see “[Modifying the Actor Port System Priority](#)” on [page 7-19](#))
- Actor port priority (see “[Modifying the Actor Port Priority](#)” on [page 7-20](#))

Note. See “[Configuring Ports to Join and Removing Ports in a Dynamic Aggregate Group](#)” on [page 7-10](#) for information on modifying a dynamic aggregate group administrative key.

Note. 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 7-21)
- Partner port administrative key (see “[Modifying the Partner Port Administrative Key](#)” on page 7-23)
- Partner port system ID (see “[Modifying the Partner Port System ID](#)” on page 7-23)
- Partner port system priority (see “[Modifying the Partner Port System Priority](#)” on page 7-24)
- Partner port administrative state (see “[Modifying the Partner Port Administrative Status](#)” on page 7-25)
- Partner port priority (see “[Modifying the Partner Port Priority](#)” on page 7-25)

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
<code>active</code>	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.
<code>timeout</code>	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.
<code>aggregate</code>	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.
<code>synchronize</code>	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.

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 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:

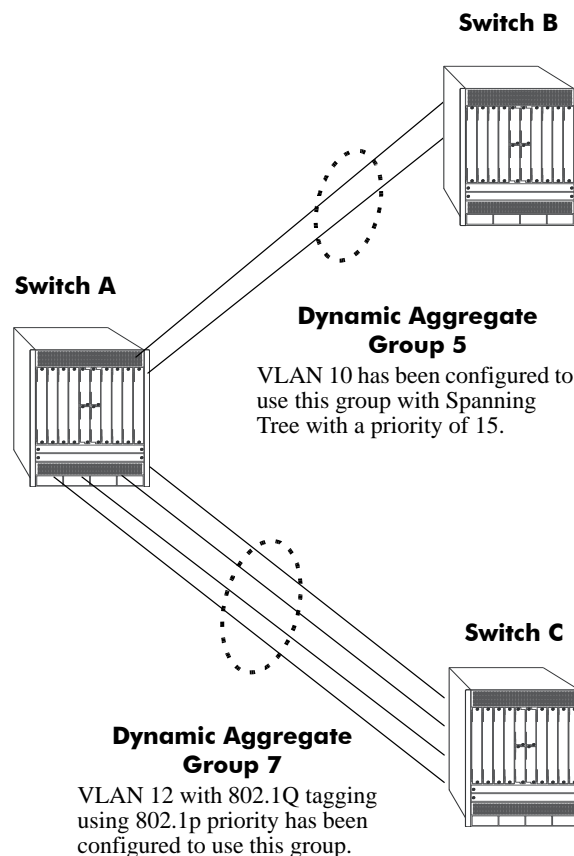
```
-> no linkagg lacp port 4/3 partner admin-port priority
```

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.



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 7-28. The steps to configure VLAN 12 (802.1Q and 802.1p example) are described in [“Link Aggregation and QoS Example”](#) on page 7-29.

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 QoS Example

As shown in the figure on [page 7-27](#), 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.aggl.syn0.col0.dis0.def1.exp0,
  Actor Oper State      : act1.tim1.aggl.syn0.col0.dis0.def1.exp0,
  Partner Admin State   : act0.tim0.aggl.syn1.col1.dis1.def1.exp0,
  Partner Oper State    : act0.tim0.aggl.syn0.col1.dis1.def1.exp0
```

Note.

When the multi-chassis link aggregation feature is activated on the switch, the **show linkagg port** command displays the output as MC-Dynamic Aggregable Port.

See the “Link Aggregation Commands” chapter in the *OmniSwitch CLI Reference Guide* for complete documentation of **show** commands for link aggregation.

8 Configuring Multi-chassis Link Aggregation

The Multi-chassis Link Aggregation feature (MC-LAG) provides resiliency at the edge of the network by enabling dual homing of any standards-based edge switches to a pair of aggregation switches to provide a Layer 2 multi-path infrastructure. MC-LAG enables a device to form a logical link aggregation (LAG) interface with two or more other devices, providing additional benefits over traditional LAG in terms of node level redundancy, multi-homing support, and loop-free Layer 2 network.

MC-LAG allows links that are physically connected to two different OmniSwitch 10K to appear as a single link aggregation group to a third edge device. MC-LAG provides this functionality without running layer 2 loop-detection protocols such as Spanning Tree Protocol between the edge and aggregation switches, while still detecting data loop conditions, failure detection and convergence.

The following are some key points regarding MC-LAG configuration:

- MC-LAG provides active/active dual homed connectivity to standards based L2 edge devices. There is no support for standby ports.
- Internal automatic configuration will disable spanning tree functionality on MC-LAG aggregate ports.
- MC-LAG peers are seen as one aggregated group to dual homed edge device(s).
- MAC addresses learned on an MC-LAG aggregate in one of the multi-chassis peers are also learned on the other switch on the same MC-LAG aggregate.
- A loop or duplicate packet prevention mechanism is implemented so that non-unicast frames received on the Virtual Fabric Link are not flooded out any local MC-LAG ports

For more information on components of MC-LAG, see [“MC-LAG Concepts and Components” on page 8-9](#)

In This Chapter

This chapter describes the basic components of MC-LAG 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 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- [“Quick Steps for Configuring MC-LAG” on page 8-5](#)
- [“MC-LAG Overview” on page 8-9](#)
- [“MC-LAG Topologies” on page 8-13](#)
- [“MC-LAG Packet Flow” on page 8-18](#)
- [“Configuring MC-LAG” on page 8-21](#)
- [“Interaction with Other Features” on page 8-27](#)
- [“MC-LAG Configuration Example” on page 8-28](#)
- [“Displaying MC-LAG Configuration and Statistics” on page 8-30](#)

Multi-chassis Link Aggregation Specifications

The table below lists specifications for dynamic aggregation groups and ports:

Maximum number of MC-LAG aggregates on multi-chassis domain	128
Maximum number of LAG aggregates on multi-chassis domain	128
Combined maximum number of MC-LAG and LAG aggregates on multi-chassis domain	128
Maximum number of ports per MC-LAG aggregate	8
Maximum number of MC-LAG peer switches	2
Valid chassis identifier	1 or 2
Maximum number of Virtual Fabric Links	1
Maximum number of ports per Virtual Fabric Link	8

Multi-chassis Link Aggregation Default Values

The table below lists default values for dynamic aggregate groups.

Parameter Description	Command	Default Value/Comments
Multi-chassis chassis ID	multi-chassis chassis-id	N/A - Not an MC-LAG peer switch.
Hello-interval	multi-chassis hello-interval	1 second
IPC-VLAN	multi-chassis ipc-vlan	4094
VLAN range on the virtual fabric	multi-chassis vf-link default-vlan	1-4094
Aggregate Identifier ranges	linkagg range local peer multi-chassis	Local: 0-47 Remote: 48-95 Multi-chassis: 96-127

Quick Steps for Configuring MC-LAG

Follow the steps below for a quick tutorial on configuring MC-LAG between two OmniSwitch 10K. Additional information on how to configure MC-LAG is provided in the section [“Configuring MC-LAG” on page 8-21](#).

Note. Although some parameters are configurable at runtime, it is strongly recommended that the entire configuration be completed prior to rebooting the switches. This will avoid a temporary mismatch of configurations between the peer switches as well as prevent multiple reboots.

1 Configure a globally unique chassis identifier using the [multi-chassis chassis-id](#) command as shown below:

```
OS10K [Chassis 1] -> multi-chassis chassis-id 1
OS10K [Chassis 2] -> multi-chassis chassis-id 2
```

2 Create a virtual fabric link between chassis peers using the [multi-chassis vf-link create](#) command as shown below:

```
OS10K [Chassis 1] -> multi-chassis vf-link create
OS10K [Chassis 2] -> multi-chassis vf-link create
```

3 Add the physical ports as members of the virtual fabric link on each peer switch using the [multi-chassis vf-link member-port](#) command as shown below:

```
OS10K [Chassis 1] -> multi-chassis vf-link member-port 1/1
OS10K [Chassis 1] -> multi-chassis vf-link member-port 1/17
OS10K [Chassis 1] -> multi-chassis vf-link member-port 3/1
OS10K [Chassis 1] -> multi-chassis vf-link member-port 3/17

OS10K [Chassis 2] -> multi-chassis vf-link member-port 2/1
OS10K [Chassis 2] -> multi-chassis vf-link member-port 2/17
OS10K [Chassis 2] -> multi-chassis vf-link member-port 4/1
OS10K [Chassis 2] -> multi-chassis vf-link member-port 4/17
```

4 Verify the chassis identifier settings using the [show multi-chassis status](#) command as shown below:

```
OS10K [Chassis 1] -> show multi-chassis status
```

```
Multi-Chassis Operational Configured
```

```
-----+-----+-----
Chassis ID      N/A          1
Chassis Role    Unassigned   N/A
Status          Standalone   N/A
Hello Interval  1s           1s
IPC VLAN        4094         4094
```

```
OS10K [Chassis 2] -> show multi-chassis status
```

```
Multi-Chassis Operational Configured
```

```
-----+-----+-----
Chassis ID      N/A          2
Chassis Role    Unassigned   N/A
```

Status	Standalone	N/A
Hello Interval	1s	1s
IPC VLAN	4094	4094

5 Verify the link aggregate identifier ranges using the **show linkagg range** command as shown below:

OS10K [Chassis 1] -> **show linkagg range**

	Operational		Configured	
	Min	Max	Min	Max
Local	0	127	0	47
Peer	N/A	N/A	48	95
Multi-Chassis	N/A	N/A	96	127

OS10K [Chassis 2] -> **show linkagg range**

	Operational		Configured	
	Min	Max	Min	Max
Local	0	127	0	47
Peer	N/A	N/A	48	95
Multi-Chassis	N/A	N/A	96	127

6 Verify the virtual fabric link configuration and default VLAN settings using the **show multi-chassis vf-link** command as shown below:

OS10K [Chassis 1] -> **show multi-chassis vf-link**

VFLink ID	Oper	Primary Port	Config Port	Active Port	Def Vlan
0	Disabled	N/A	0	0	1

OS10K [Chassis 2] -> **show multi-chassis vf-link**

VFLink ID	Oper	Primary Port	Config Port	Active Port	Def Vlan
0	Disabled	N/A	0	0	1

7 Verify the virtual fabric link configuration using the **show multi-chassis vf-link member-port** command as shown below:

OS10K [Chassis 1] -> **show multi-chassis vf-link member-port**

VFLink ID	Slot/Port	Oper	Is Primary
0	1/1	Disabled	No
0	1/17	Disabled	No
0	3/1	Disabled	No
0	3/17	Disabled	No

OS10K [Chassis 2] -> **show multi-chassis vf-link member-port**

VFLink ID	Slot/Port	Oper	Is Primary
0	2/1	Disabled	No
0	2/17	Disabled	No
0	4/1	Disabled	No
0	4/17	Disabled	No

8 Verify the consistency of system-level mandatory parameters between the two chassis using the **show multi-chassis consistency** command as shown below:

OS10K [Chassis 1] -> **show multi-chassis consistency**

Consistency	Local	Peer	Status
Chassis-ID	N/A	N/A	NOK
Hello-Interval	1	N/A	NOK
IPC-VLAN	4094	N/A	NOK
STP-Path-Cost-Mode	Auto	N/A	NOK
STP-Mode	Per-VLAN	N/A	NOK

OS10K [Chassis 2] -> **show multi-chassis consistency**

Consistency	Local	Peer	Status
Chassis-ID	N/A	N/A	NOK
Hello-Interval	1	N/A	NOK
IPC-VLAN	4094	N/A	NOK
STP-Path-Cost-Mode	Auto	N/A	NOK
STP-Mode	Per-VLAN	N/A	NOK

9 Save the configuration and reload using the **write memory** and **reload** commands as shown below:

```
-> write memory
-> reload from working no rollback-timeout
```

10 Once the system reboots verify the multi-chassis functionality using the commands shown below:

OS10K [Chassis 1] -> **show multi-chassis vf-link member-port**

VFLink ID	Slot/Port	Oper	Is Primary
0	1/1	Up	Yes
0	1/17	Up	No
0	3/1	Up	No
0	3/17	Up	No

OS10K [Chassis 2] -> **show multi-chassis vf-link member-port**

VFLink ID	Slot/Port	Oper	Is Primary
0	2/1	Up	No
0	2/17	Up	Yes
0	4/1	Up	No
0	4/17	Up	No

OS10K [Chassis 1] -> **show multi-chassis status**

Multi-Chassis	Operational	Configured
Chassis ID	1	1
Chassis Role	Primary	N/A
Status	Up	N/A
Hello Interval	1s	1s
IPC VLAN	4904	4094

OS10K [Chassis 2] -> **show multi-chassis status**

Multi-Chassis	Operational	Configured
Chassis ID	2	2
Chassis Role	Secondary	N/A
Status	Up	N/A
Hello Interval	1s	1s
IPC VLAN	4904	4094

OS10K [Chassis 1] -> **show multi-chassis consistency**

Consistency	Local	Peer	Status
Chassis-ID	1	2	OK
Hello-Interval	1	1	OK
IPC-VLAN	4094	4094	OK
STP-Path-Cost-Mode	Auto	Auto	OK
STP-Mode	Per-VLAN	Per-VLAN	OK

OS10K [Chassis 2] -> **show multi-chassis consistency**

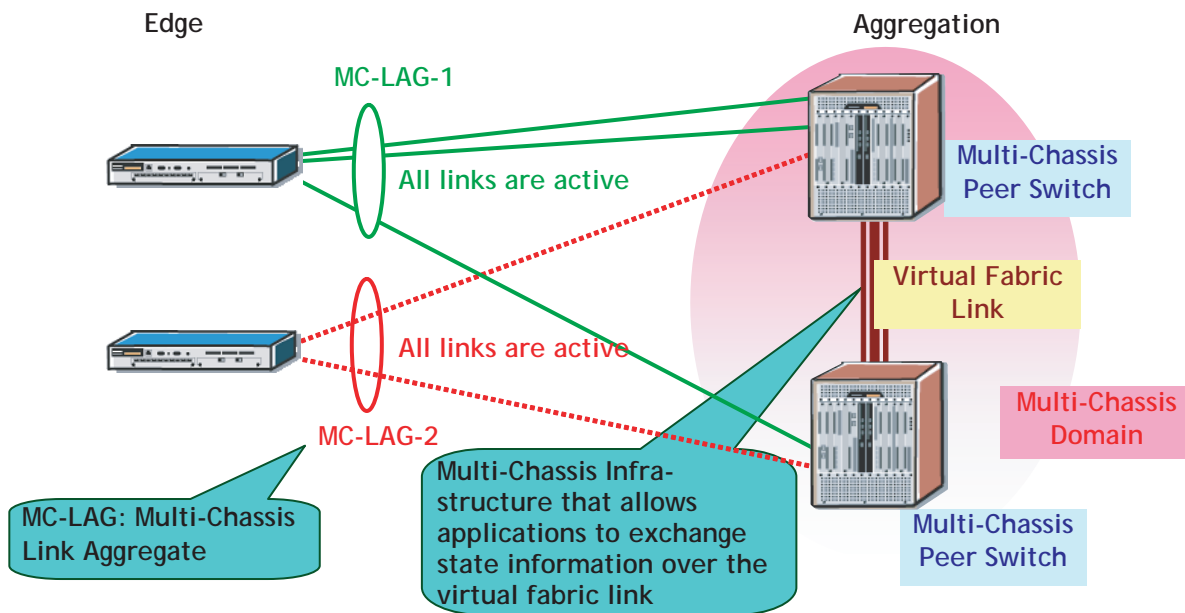
Consistency	Local	Peer	Status
Chassis-ID	2	1	OK
Hello-Interval	1	1	OK
IPC-VLAN	4094	4094	OK
STP-Path-Cost-Mode	Auto	Auto	OK
STP-Mode	Per-VLAN	Per-VLAN	OK

Note. Multi chassis peers in the same domain must maintain identical configuration and operational parameters. Ensure that the mandatory parameters are the same on both peers. For more information, see [“Recommended Configuration Parameters” on page 8-25](#)

MC-LAG Overview

Multi Chassis Link Aggregation (MC-LAG) addresses resiliency at the edge of the network by enabling dual homing of any standards-based edge switches to a pair of aggregation switches resulting in a Layer 2 multi-path infrastructure. MC-LAG provides increased bandwidth, load balancing and resiliency for L2 edge devices in a network.

An edge switch is dual homed to two Omniswitch10K through Link Aggregation Control Protocol or Static Aggregation. MC-LAG enhances link aggregation by eliminating blocked redundant links to provide fast switch over between edge and core switches without implementing Spanning Tree. Dual homed uplinks are active /active and provide sub second traffic convergence for link fail.



Example of a MC-LAG Group Network

MC-LAG Concepts and Components

MC-LAG is an OmniSwitch 10K feature that requires complex building blocks to provide full functionality. The following sections highlight various aspects of MC-LAG.

The Multi chassis domain is a virtual entity consisting of a peer OmniSwitch 10K, the virtual fabric link, all of the MC-LAG aggregate ports and the edge devices attached to MC-LAGs. A Multi-Chassis domain can support up to 128 MC-LAG groups.

Multi-Chassis Manager (MCM) is an Alcatel Lucent proprietary application that manages and monitors the multichassis functionality. MCM provides the foundation for inter-chassis communication infrastructure that can be used by applications (e.g. link aggregation) to exchange state information.

- MCM Manages and monitors multi-chassis functionality, performs peer discovery and establishes the VFL.
- MCM performs and monitors consistency parameters between peer chassis and runs the hello protocol between peer chassis.

Edge Switches are any L2 standards-based switches providing network access to client PCs, servers, printers, etc. These devices dual home (active/active) into MC-LAG groups aggregated across a pair of OmniSwitch 10K multi-chassis peers.

Multi-Chassis Peer Switches are switches that terminate the aggregate links coming from multiple edge switches; two switches are supported per MC-LAG domain. Each peer member can be a member of only one domain and must be assigned with a unique chassis ID. MC-LAG functionality will remain operationally down if duplicate chassis IDs are detected. The primary and secondary roles are only effective when both chassis are up and MC-LAG is operational.

IPC VLAN is a special VLAN reserved for inter-chassis communication exchange between multi-chassis peers. Ensure that IPC VLAN is the same on both peers. Only VFL ports can be assigned to this VLAN and no other ports are allowed to join the IPC VLAN.

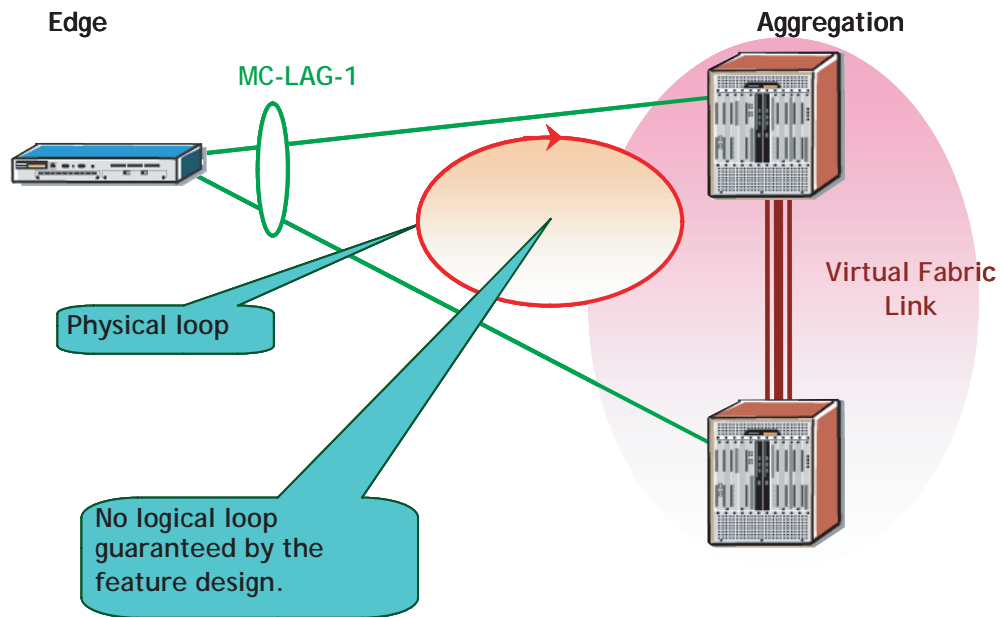
Virtual Fabric Link (VFL) is the aggregated group of 10G ports interconnecting the multi-chassis peers. VFL is an automatic member of all VLANs created on the local chassis. The operational state of VFL is tracked by implementing Link Aggregation Control Protocol. Group ID 128 is reserved for VFL aggregate ports on single or multiple slots.

Virtual IP Interface (VIP) is the IP interface bound to the VFL for the respective subnet. It is the default gateway for devices attached to the respective subnet. The VIP cannot be bound to any routing protocol or establish any routing adjacencies. It routes only locally between MC-LAG subnets. By configuring the routing maps, VIP subnets can converge with external routes.

Loop Detection is a utility provided to enable network loop detection. As a rule, you must take care not to introduce back door loops through edge devices. The loop detection is enabled or disabled per system. Loop detection mechanism generates multicast Loop Detect PDU at regular interval.

Benefits of MC-LAG

- MC-LAG ensures high-availability in the network by providing node resiliency on the aggregation layer.
- MC-LAG provides dual-homed Layer 2 multi-path connections for edge nodes into the aggregation without running the Spanning Tree protocol. The edge device can be any LACP capable-device.
- MC-LAG delivers active/active forwarding mode whereby both sets of uplinks that are part of the dual homed aggregates are processing traffic to maximize the value of the customer investment.



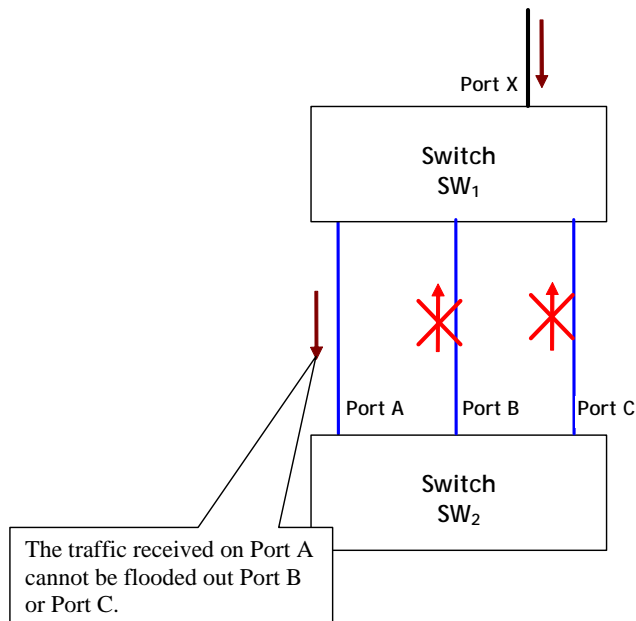
Example of a MC-LAG Group Network

An important characteristic of this solution relates to the absence of a logical loop between the edge and multi-chassis peer switches, even though a physical loop does exist.

MC-LAG Principle

In order to ensure a loop free topology, traffic received on one of the ports of an aggregate is never flooded out to any of the member ports including the receiving port or through the VFL.

In the figure below assume 2 switches connected back to back through an LACP interface. If flooded traffic (multicast, broadcast, unknown traffic) is received by switch SW2 on Port A, it will not be flooded out on either Port B or Port C. Additionally, it cannot be sent back out of the same port where it was received.



Basic MC-LAG Principle

MC-LAG Loop Detection

Since the Spanning Tree Protocol is disabled on MC-LAG ports, the MC-LAG feature provides a method of loop detection in order to detect an invalid network topology. The mechanism provides loop detection for any potential loops that include a set of MC-LAG aggregate ports.

The loop detection mechanism generates multicast Loop Detect PDU at regular intervals. In a MC-LAG network, the source MAC is reserved and the MAC is unique to each chassis ID. The multicast PDU is flooded out on the VFL and MC-LAG primary ports.

Loop Detection is flagged when the PDU is returned to the transmitting peer, causing the following to occur.

- A log message is sent for loop detect event.
- A SNMP trap is generated, and
- The offending port is shutdown.

MC-LAG Topologies

This section describes the building blocks that are used to construct more flexible and complex network topologies. The recommended topologies for MC-LAG that are supported are:

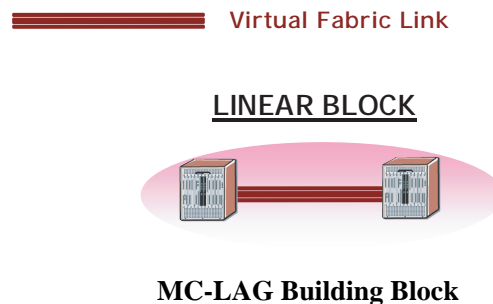
- When all edge devices are attached to both MC-LAG peers at the core.
- When edge switches are connected through MC-LAG and core switches are dual-attached to non MC-LAG interfaces.

For more information on MC-LAG topologies, refer to the following sections.

- [“Basic MC-LAG Building Block” on page 8-13](#)
- [“Recommended Topologies” on page 8-13](#)
- [“Topologies Not Recommended” on page 8-15](#)
- [“Unsupported Topologies” on page 8-15](#)

Basic MC-LAG Building Block

The following diagram illustrates the basic building block that can be used to construct more flexible and complex network topologies. The building block below can be used to connect to the edge or core devices in the network and is comprised of two OmniSwitch 10K chassis with a VFL link configured between them.



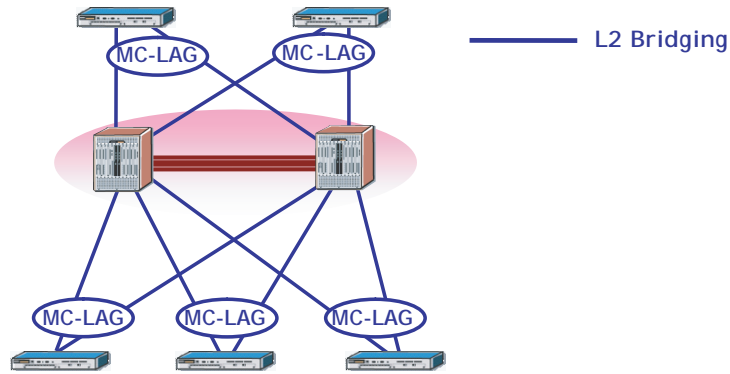
Recommended Topologies

The following topologies are recommended to support MC-LAG:

- MC-LAG at L2 Core
- MC-LAG at Aggregation Layer

MC-LAG at the L2 Core

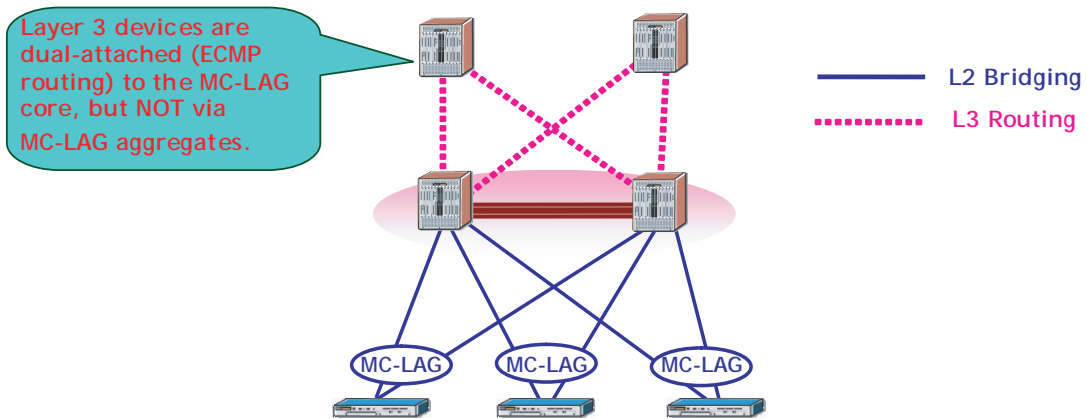
In the topology shown below, all edge devices are attached to both MC-LAG peers at the core. Spanning Tree is not needed in this network because there are no loops. In this topology, the physical loop around the MC-LAG ports and Virtual Fabric Link is prevented by the MC-LAG.



MC-LAG at the L2 Core

MC-LAG at the Aggregation Layer

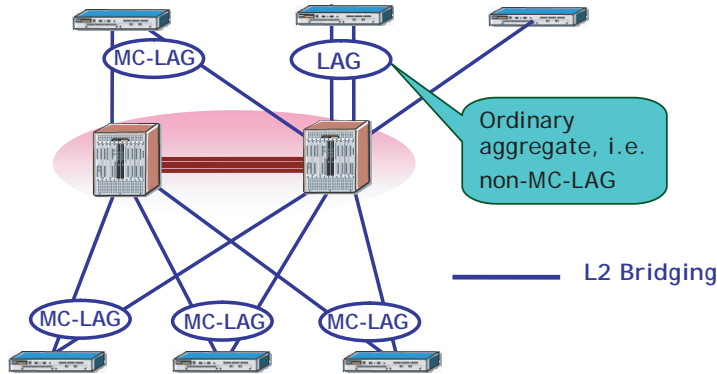
In the topology shown below, edge switches are connected through MC-LAG and core switches are dual attached.



MC-LAG at the Aggregation Layer

Topologies Not Recommended

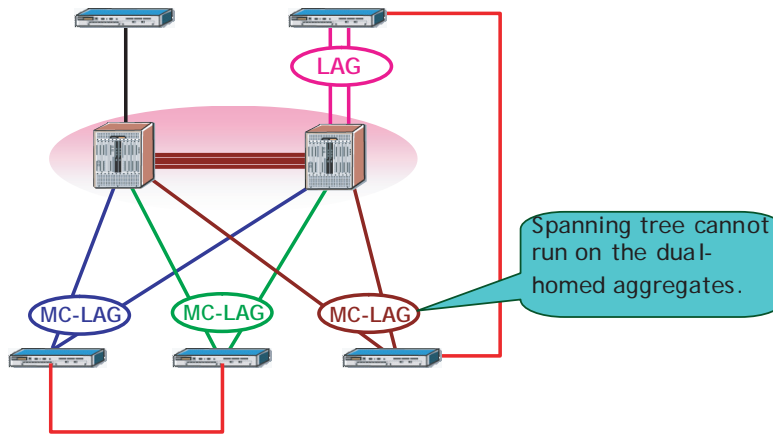
In the topology shown below, edge devices are not attached to both MC-LAG peers at the core. As a result, some traffic may need to constantly flow across the Virtual Fabric Link. Still, Spanning Tree is not required as there are no logical loops in this network.



Edge Switches Without MC-LAG

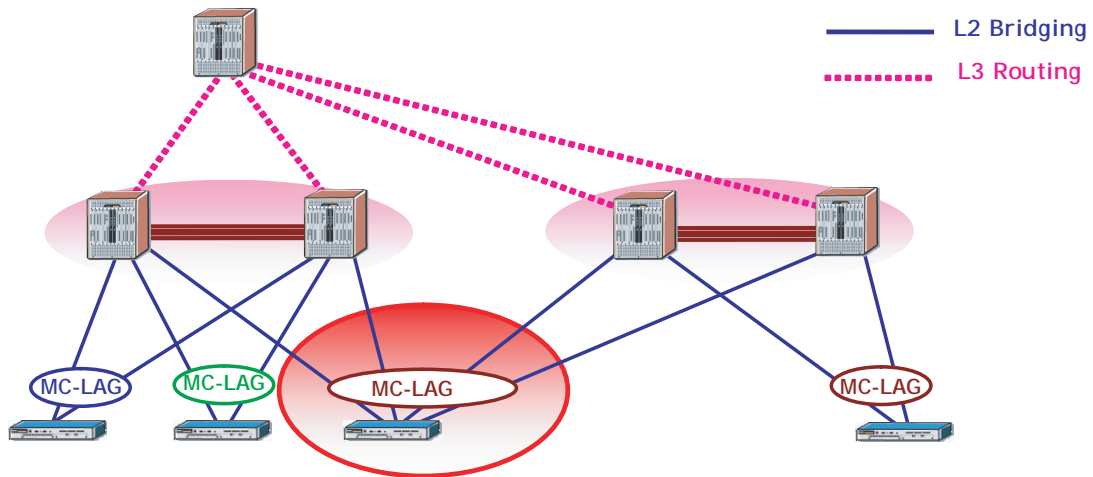
Unsupported Topologies

In the topology shown below, MC-LAG is not supported since Spanning Tree cannot run with a “back-door” connection and will result in a loop.



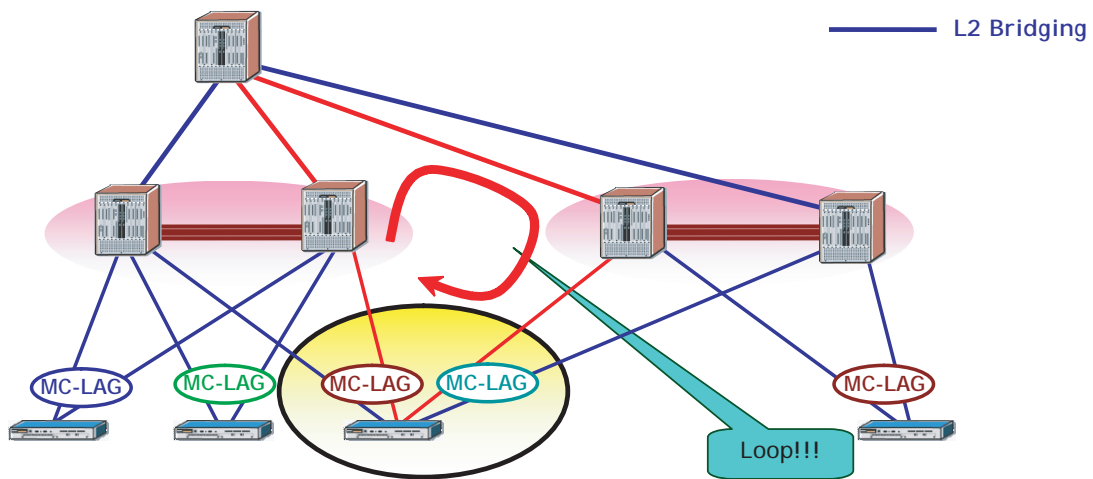
Back-door Connection Causing Physical Loop

This topology violates the principle that each edge switch can only be part of a single MC-LAG domain and increases the configuration complexity. Now all the four multi-chassis peers need to have consistent configurations (for example, LACP System ID in order for the edge switch to be able to negotiate the four links as part of the same aggregate).



Edge Switch to Multiple MC-LAG Domains

This topology introduces the risk of a possible loop indicated by the arrows. Since Spanning Tree will not run over the MC-LAG aggregates, this loop cannot be prevented.

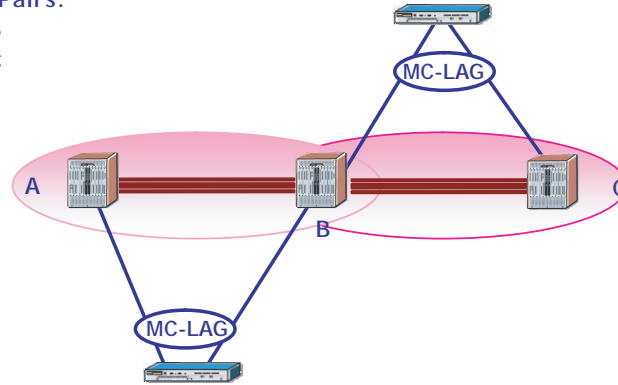


Edge Switch to Multiple MC-LAG Domains

The following topology illustrates that Switch B is required to keep separate system resources, such as MAC tables, ports, software applications per virtual domain.

MC-LAG Pairs:

- A - B
- B - C

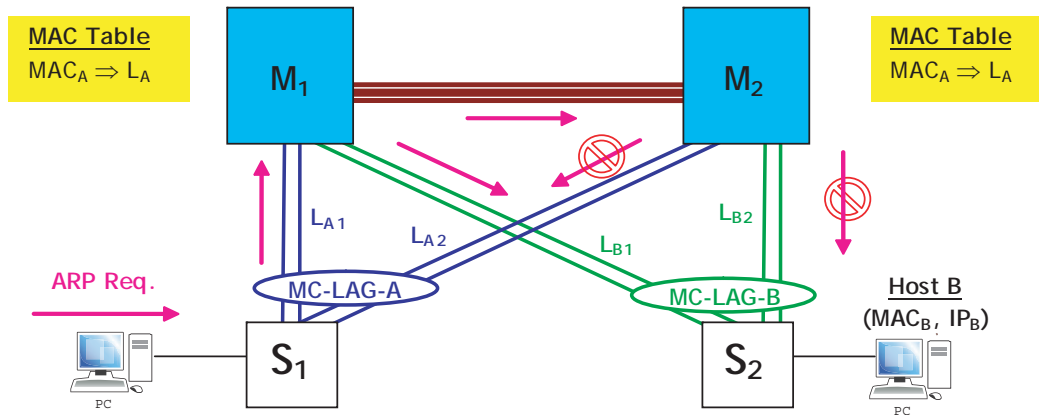


Overlapping MC-LAG Switch Pairs

MC-LAG Packet Flow

Layer 2 Switching over MC-LAG

Since hosts A and B are within the same IP subnet and VLAN, host A has a directly connected route to reach B through the outgoing interface connected to switch S1. Host A needs to determine the MAC address of host B.



ARP Request Over MC-LAG

1 Since both hosts are in the same IP subnet, host A will send an ARP request as follows:

- Source MAC = MAC_A
- Destination MAC = ff:ff:ff:ff:ff:ff (Broadcast)
- Target IP = IP_B

Depending on the hash algorithm or use of a fixed primary port for non-unicast traffic, switch S₁ will select a different port of the aggregate MC-LAG A to send the ARP request. In this example, assume that the request goes through one of the ports connected to M1 represented by L_{A1}.

2 Loop Prevention

- The broadcast packet is simply flooded within the system as indicated by the arrows.
- The MC-LAG will prevent the flooded packets received by M₂ through the Virtual Fabric Link from being sent out its local MC-LAG ports.
- This way, S₂ will not get duplicate copies of the original packet that would otherwise flow through two distinct paths: S₁ ==> M₁ ==> S₂ and S₁ ==> M₁ ==> M₂ ==> S₂.

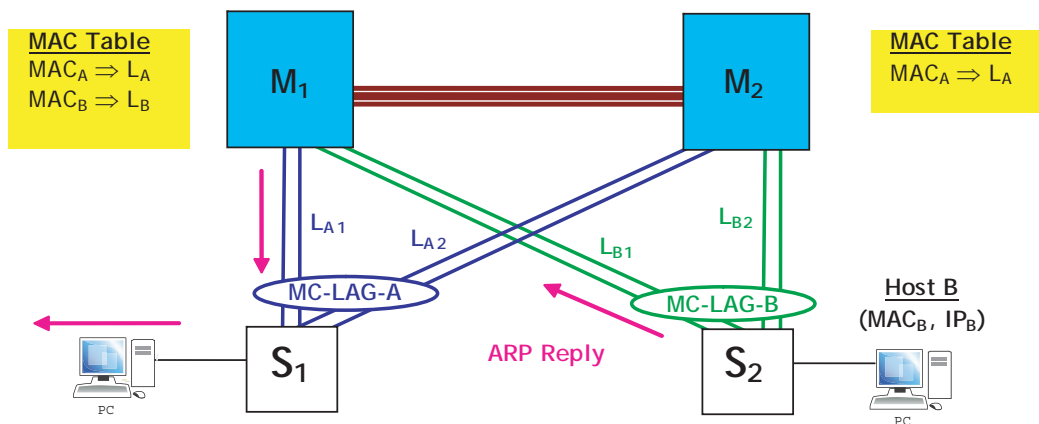
3 Step 3: MAC_A Learning

- Switch M₁ will learn MAC@ = MAC_A on the MC-LAG aggregate L_A.
- Switch M₂ will learn MAC@ = MAC_A on the MC-LAG aggregate L_A as well.

MC-LAG prevents MAC_A from being learned on the VFL ports of switch M_2 but shows MAC_A as learned on the L_A aggregate even though the original ARP request packet was actually received through the VFL. The MAC addresses learned on MC-LAG ports in one switch are learned on the same MC-LAG ports on the peer switch.

4 Step 4: ARP Reply

- Switches M_1 and M_2 do not respond to the ARP request because the destination IP address (IP_B) is not an IP address of any of either of their IP interfaces.
- Eventually, the ARP request will reach the destination host B, which will respond to the ARP request as indicated below.



ARP Reply Over MC-LAG

The ARP reply is a unicast packet as follows.

- Source MAC = MAC_B
- Destination MAC = MAC_A

5 Step 5: MAC_B Learning

As the ARP reply packet traverses the system on its way back via the path Host B \Rightarrow $S_2 \Rightarrow M_1 \Rightarrow S_1 \Rightarrow$ Host A, the MAC_B address is learned by the M_1 switch on the aggregate L_B .

6 Step 6: Regular Traffic Flow

The packet eventually reaches the Host A. From this moment on, Host A will be able to communicate with Host B using IP over the Layer 2 multi-path infrastructure provided by the MC-LAG aggregates.

Key Points

- The MC-LAG peers perform only Layer 2 switching and learning operations despite the fact that actual traffic flowing between the hosts is IP-based.
- MAC addresses learned on an MC-LAG aggregate on one of the MC-LAG peers are also learned on the other peer on the same MC-LAG aggregate.

-
- A loop/duplicate packet prevention mechanism is implemented so that non-unicast frames received on the Virtual Fabric Link are not flooded out any local MC-LAG ports.
 - Downstream traffic always prefers the local MC-LAG ports, if these are available.

Configuring MC-LAG

This section describes commands to configure MC-LAG on a switch.

- [“MC-LAG Configuration Guidelines” on page 8-21](#)
- [“Configuring the Chassis-ID” on page 8-23](#)
- [“Configuring the IPC-VLAN” on page 8-23](#)
- [“Configuring the Hello-Interval” on page 8-23](#)
- [“Configuring Aggregate Identifier Ranges” on page 8-23](#)
- [“Creating the Virtual Fabric Link \(VFL\)” on page 8-24](#)
- [“Configuring MC-LAG Aggregates” on page 8-24](#)
- [“Configuring the VIP VLAN” on page 8-24](#)

Note. See [“Quick Steps for Configuring MC-LAG” on page 8-5](#) for a brief tutorial on configuring these parameters on an OmniSwitch 10K.

MC-LAG Configuration Guidelines

The following sections provide configuration guidelines to follow when configuring MC-LAG on an OmniSwitch 10K.

General

- A VLAN that performs IP routing cannot span several switches throughout the entire network if any of its member ports is a multi-chassis aggregate.
- The Spanning Tree protocol can run on MC-LAG chassis peers, however STP is disabled on MC-LAG ports.
- One of the MC-LAG chassis peers should be the root bridge so that the VFL is always in forwarding mode.
- Due to the MG-LAG loop avoidance feature, non-unicast traffic received on the VFL is never flooded on local MC-LAG ports.
- There is no synchronization of routing information between MC-LAG peers.

Chassis-ID

- Each chassis requires a globally unique chassis ID within the allowed range (1-2).
- If a duplicate chassis ID is detected, then the operational state of the chassis will remain down.
- Chassis ID information is distributed to all NIs to allow NI software to program the underlying ASICs with the correct module ID derived from the chassis ID.
- The switch must be rebooted after configuring the chassis ID.

For information about configuring the Chassis-ID, see [“Configuring the Chassis-ID” on page 8-23](#)

Virtual Fabric Link

- The operational state of the multi-chassis functionality depends on the VFL operational state. You must explicitly configure the VFL and specify the physical port members.
- VFL should be configured only during the network maintenance or during the initial MC-LAG configuration. Changing VFL configuration at runtime can cause undesirable disruption to traffic flows.
- LACP Aggregate ID 128 is reserved and assigned to VFL. For increased resiliency, member ports should be distributed across different switching ASICs and NI modules
- It is recommended to configure the VFL at the same time as the chassis identifier. This ensures that the switch reboots with the correct VFL configuration.
- Only physical ports operating at 10-Gbps can be members of a VFL. The VFL is automatically a member of all VLANs.
- All the member ports must operate at the same speed. If more than one member port is configured, they will be bundled to form a single logical link.
- The hello interval parameter must match between chassis peers. The hello protocol runs across the VFL link between the peers.

For more information on Virtual Fabric Link, see [“Creating the Virtual Fabric Link \(VFL\)” on page 8-24](#)

IPC VLAN

- The IPC VLAN ID must match between the two chassis and cannot be disabled.
- Only the VFL link can be configured as a member of the IPC VLAN.

For more information on IPC-VLAN, see [“Configuring the IPC-VLAN” on page 8-23](#)

Aggregate Range Identifiers

- The switch must be rebooted after configuring Aggregate Range Identified values.
- The local range configured on Chassis 1 must match the peer range configured on Chassis 2 and vice-versa.
- Only an aggregate ID in the Multi-chassis range should be used for MC-LAG.

For more information about Aggregate Identifier Ranges, see [“Configuring Aggregate Identifier Ranges” on page 8-23](#)

VIP VLAN

- A chassis identifier must already be configured on the OmniSwitch 10K.
- The only ports that can be configured as members of a VIP VLAN are the multi-chassis link aggregates.
- Routing protocols cannot be configured on IP interfaces assigned to a VIP VLAN.
- VRRP cannot be configured on IP interfaces assigned to a VIP VLAN.
- VIP VLANs must have both a management IP and a virtual IP address.

For more information on VIP VLAN, see [“Configuring the VIP VLAN” on page 8-24](#)

Configuring the Chassis-ID

To configure MC-LAG, a globally unique chassis identifier must first be assigned to each of the switches that will form the multi-chassis domain. The chassis ID is used to generate globally unique values for the module identifiers as well as allowing inter-chassis communication.

To configure an OmniSwitch 10K for MC-LAG and assign a globally unique chassis identifier, enter the **multi-chassis chassis-id** command as shown below:

```
-> multi-chassis chassis-id 1
```

By default, the MC-LAG hello interval and IPC VLAN are set to the values listed in “[Multi-chassis Link Aggregation Default Values](#)” on page 8-4 and do not need to be modified. See “[Configuring the IPC-VLAN](#)” on page 8-23 and “[Configuring the Hello-Interval](#)” on page 8-23 for additional information.

Configuring the IPC-VLAN

Under normal circumstances, it is not necessary to change the IPC VLAN listed in the Default Value table at the beginning of the chapter.

However, it is important to note that the VLAN configured as the IPC-VLAN is reserved specifically for MC-LAG purposes and can no longer be used for normal data traffic.

If necessary, use the **multi-chassis ipc-vlan** command to modify the IPC VLAN as show below:

```
-> multi-chassis ipc-vlan 4093
```

Configuring the Hello-Interval

Hello packets are used for establishing and maintaining the neighbor relationship between multi-chassis peers and ensures that communication between peers is bi-directional. Hello packets are sent periodically out VFL interfaces. Bidirectional communication is indicated when the switch sees itself listed in the neighbor's Hello Packet.

To configure the hello interval between the multi-chassis peers, use the **multi-chassis hello-interval** command as shown below:

```
-> multi-chassis hello-interval 1
```

Configuring Aggregate Identifier Ranges

The aggregate identifier ranges are the valid ranges defined for ordinary aggregates as well as the multi-chassis link aggregates. Although the default values will typically suffice, these values can be modified, if necessary, using the **linkagg range local peer multi-chassis** command as shown below:

```
-> linkagg range local 0-47 peer 48-95 multi-chassis 96-127
```

Note: The local range configured on Chassis 1 must match the peer range configured on Chassis 2 and vice-versa.

Creating the Virtual Fabric Link (VFL)

The VFL is an aggregate of high-speed ports used for inter-chassis traffic and control data through the IPC-VLAN. For MC-LAG to become operational, a VFL must be configured and brought to an operational state.

To configure a VFL and its member ports, enter the **multi-chassis vf-link create** command as shown below:

```
-> multi-chassis vf-link
-> multi-chassis vf-link member-port 2/1
-> multi-chassis vf-link member-port 2/2
```

Configuring the VFL Default VLAN

Traffic belonging to the VFL default VLAN will be sent across the VFL untagged. The VFL will also be a member of any additional VLANs configured on the MC-LAG chassis peers as tagged VLANs.

To configure the VFL default VLAN, enter the **multi-chassis vf-link default-vlan** command as shown below:

```
-> multi-chassis vf-link default-vlan 2
```

Configuring MC-LAG Aggregates

MC-LAG aggregates can be configured using either static or dynamic link aggregation. The key point when configuring the aggregates is that from the edge switch's point of view, it looks like the edge is connected to a single chassis.

Configuring the VIP VLAN

A VIP VLAN is configured when routing is to be performed on the OmniSwitch 10K. A VIP VLAN has two IP addresses associated with it:

- **Management address** - A unique IP address used by each switch within a multi-chassis system to provide management services.
- **Virtual IP address** - Used for supporting routing of packets terminated on the multi-chassis switches. Unlike the management address, the virtual IP address must be the same on the peer switches.

To configure a switch for MC-LAG and assign a globally unique chassis identifier, enter the **ip interface** command as shown below:

```
-> ip interface vip-vlan-10 address 10.10.10.1/24 vip-address 10.10.10.100 vlan
10
```

Recommended Configuration Parameters

Multi-chassis peers in the same domain must maintain identical configuration and operational parameters. Ensure that the mandatory parameters are the same on both peers. Multi-chassis functionality is optimal with unified management on participating chassis. Any mismatch or mis-configuration can adversely affect network traffic behavior. There is no automatic check or facility to check for mis-configuration.

Ensure that the following recommended parameters are configured identical on both the chassis.

- MAC Address Aging Timer
- MC-LAG member ports: port speed and duplex
- Static MAC Entries
- QoS Configuration
- Port Security Configuration
- IGMP Snooping Configuration
- PIM Configuration
- IP interface configuration
- Routing Protocols Configuration
- VRRP Configuration

Verifying Parameter Consistency

Parameter consistency is performed automatically at start time and relevant runtime when any configuration changes are applied to either chassis. If there is any persistent mismatch of the mandatory parameters, then the global operation state of the Multi-Chassis feature will be updated as operationally down. Inconsistency between some of the non-mandatory MC-LAG aggregate parameters prevents a particular MC-LAG or all MC-LAG aggregates from becoming operational. The mandatory parameters for MC-LAG and the impact of their violation are: as follows:

Parameter	Violation Impact
Global Parameters	
Chassis ID (must be different between chassis)	Bring down all MC-LAG aggregates
Multi-Chassis Hello Interval	
STP Path Cost Mode: Auto, 32-bit	
STP Mode: 1x1, Flat	
Per MC-LAG Parameters	
MC-LAG LACP Type: MC-LACP, MC-Static	Bring the specific MC-LAG aggregate down
VLAN Configured on MC-LAG aggregate	
VLAN Type: Default, 802.1Q Tag	
VLAN Enable State on MC-LAG	
LACP System ID (*)	
LACP System Priority (*)	

Note. AOS has global default values for the LACP System ID (derived from the system MAC address) and LACP System Priority (a constant hard-coded value). Even though not widely used, the management interface provides the ability to change these parameters on a per-aggregate basis. As a result, these parameters are always treated as per-MC-LAG aggregate.

Interaction with Other Features

This section contains important information about how other OmniSwitch features interact with MC-LAG instances. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature. MC-LAG interaction with other features like:

Spanning Tree Protocol

Spanning Tree protocol is not allowed on logical dual-homed MC-LAG aggregates and on local MC-LAG aggregates. Spanning Tree must be disabled on the local MC-LAG aggregates within each multi-chassis peer. This is true regardless of the presence of active ports on both peer switches. MC-LAG provides an alternative to Spanning Tree while supporting dual-homed connections between the edge and aggregation switches.

Since STP is disabled on MC-LAG ports, multi-chassis link aggregation provides a method of loop detection in order to detect an invalid network topology. The mechanism provides loop detection for any potential loops that include a set of MC-LAG aggregate ports. For more information, see [“MC-LAG Loop Detection” on page 8-12](#).

Ethernet Ring Protection

Ethernet Ring Protection (ERP) is not allowed to be configured on MC-LAG aggregates. ERP can be configured on the virtual fabric link so that ERP packets can flow on this link. MC-LAG prevents VFL from going to a blocking state on either side of the link regardless of the network topology when the VFL is part of the ERP ring. In this case, if the VFL is ever blocked due to ERP convergence, a trap is generated and switch log is recorded and informed.

Link Aggregation

MC-LAG ports configured as dual-homed MC-LAG will not support standby ports.

Multicast

DVMRP is not supported with MC-LAG.

Source Learning

MC-LAG can be enabled only if the network operates in centralized source learning mode.

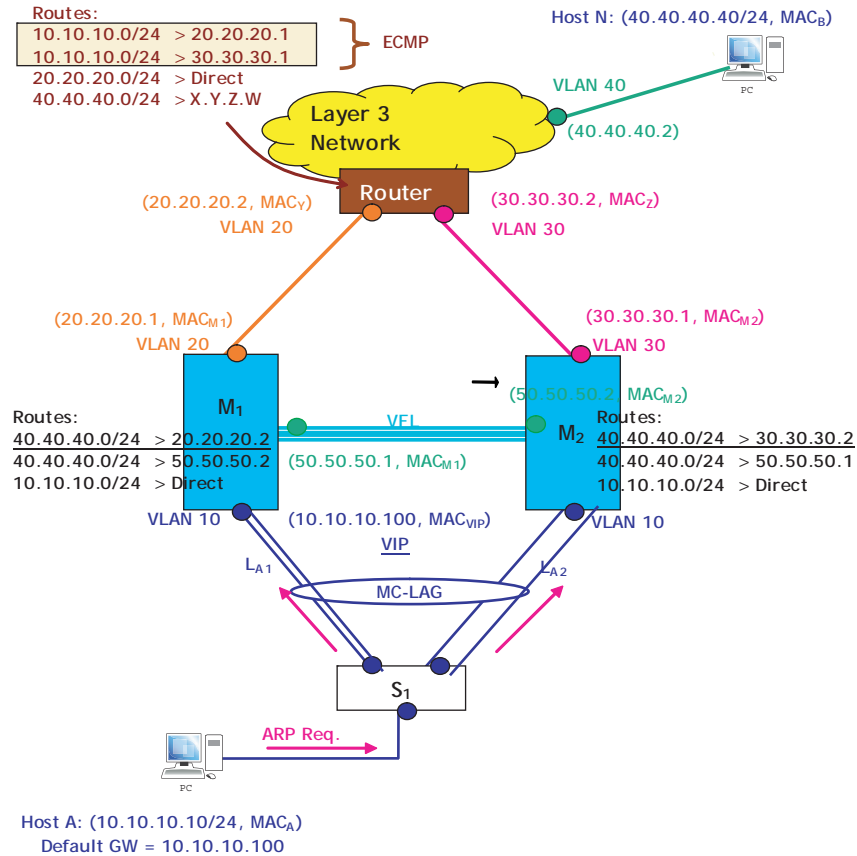
IPv4

IPv4 interfaces are supported on top of MC-LAG aggregates only through the use of a new type of VLAN called Multi-Chassis Virtual IP VLANs. In other words, IPv4 interfaces can only be configured on a VLAN that has a MC-LAG aggregate as a member of a “Multi-Chassis Virtual IP VLAN”.

Multi-Chassis Virtual IP VLANs can only be configured on MC-LAG aggregates but not on any fixed port or ordinary aggregate. However, they will always be automatically configured on the VFL link just like all other VLAN types.

MC-LAG Configuration Example

This section provides an example of MC-LAG configuration in a network.



MC-LAG L2/L3 Example

VLAN Configuration M1, M2 and S1

```

OS10K [M1] -> vlan 20
OS10K [M1] -> vlan 50
OS10K [M1] -> ip interface vlan-20 address 20.20.20.1/24 vlan 20
OS10K [M1] -> ip interface vlan-50 address 50.50.50.1/24 vlan 50
[Configure appropriate routing protocol on VLANs 20 and 50]

OS10K [M2] -> vlan 30
OS10K [M2] -> vlan 50
OS10K [M2] -> ip interface vlan-30 address 30.30.30.1/24 vlan 30
OS10K [M2] -> ip interface vlan-50 address 50.50.50.1/24 vlan 50
[Configure appropriate routing protocol on VLANs 30 and 50]

OS10K [S1] -> vlan 10
    
```

MC-LAG Configuration - M1

```
OS10K [M1] -> multi-chassis chassis-id 1

OS10K [M1] -> multi-chassis vf-link create
OS10K [M1] -> multi-chassis vf-link member-port 8/1
OS10K [M1] -> multi-chassis vf-link member-port 8/17

OS10K [M1] -> multi-chassis vip-vlan 10
OS10K [M1] -> ip interface vip-vlan-10 address 10.10.10.1/24 vip-address
10.10.10.100 vlan 10

OS10K [M1] -> write memory
OS10K [M1] -> reload working no rollback-timeout

OS10K [M1] -> linkagg lacp agg 96 size 2 admin-state enable multi-chassis active
OS10K [M1] -> linkagg lacp agg 96 actor system-id 00:00:00:00:00:01
OS10K [M1] -> linkagg lacp port 1/1-2 actor admin-key 1
OS10K [M1] -> linkagg lacp port 1/1-2 actor system-id 00:00:00:00:00:01
OS10K [M1] -> vlan 10 members linkagg 1 untagged
```

MC-LAG Configuration - M2

```
OS10K [M2] -> multi-chassis chassis-id 2

OS10K [M2] -> multi-chassis vf-link create
OS10K [M2] -> multi-chassis vf-link member-port 8/1
OS10K [M2] -> multi-chassis vf-link member-port 8/17

OS10K [M2] -> multi-chassis vip-vlan 10
OS10K [M2] -> ip interface vip-vlan-10 address 10.10.10.2/24 vip-address
10.10.10.100 vlan 10

OS10K [M2] -> write memory
OS10K [M2] -> reload working no rollback-timeout

OS10K [M2] -> linkagg lacp agg 96 size 2 admin-state enable multi-chassis active
OS10K [M2] -> linkagg lacp agg 96 actor system-id 00:00:00:00:00:01
OS10K [M2] -> linkagg lacp port 1/3-4 actor admin-key 1
OS10K [M2] -> linkagg lacp port 1/3-4 actor system-id 00:00:00:00:00:01
OS10K [M2] -> vlan 10 members linkagg 1 untagged
```

Verify MC-LAG Configuration - M1 and M2

```
OS10K [M2] -> show multi-chassis status
OS10K [M2] -> show multi-chassis consistency
OS10K [M2] -> show multi-chassis vf-link
OS10K [M2] -> show multi-chassis vf-link member-port
```

SW1 Configuration

```
OS10K [S1] -> linkagg lacp agg 96 size 4 admin-state enable
OS10K [S1] -> linkagg lacp port 1/1-2 actor admin-key 1
OS10K [S1] -> linkagg lacp port 1/3-4 actor admin-key 1
OS10K [S1] -> vlan 10 members linkagg 1 untagged
```

Displaying MC-LAG Configuration and Statistics

You can use Command Line Interface (CLI) **show** commands to display the current configuration and statistics of Multi-chassis link aggregation. These commands include the following:

show multi-chassis status	Displays the configured and operational parameters related to the multi-chassis feature on the local chassis.
show multi-chassis vf-link	Displays the configured and operational parameters related to the virtual fabric link on the local chassis.
show multi-chassis vf-link member-port	Displays the configured and operational parameters related to the virtual fabric link member ports on the local chassis
show multi-chassis consistency	Displays the system level mandatory consistency parameters of both the local and peer chassis
show multi-chassis consistency linkagg	Displays the per-multi-chassis aggregate consistency parameters of both the local and peer chassis given the aggregate identifier.

For more information about the output details that result from these commands, see the *OmniSwitch CLI Reference Guide*.

9 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.

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 CLI Reference Guide*.

The following information and configuration procedures are included in this chapter include:

- [“ERP Overview” on page 9-3.](#)
- [“Interaction With Other Features” on page 9-8.](#)
- [“ERP Configuration Overview and Guidelines” on page 9-11.](#)
- [“Configuring an ERP Ring” on page 9-12.](#)
- [“Sample Ethernet Ring Protection Configuration” on page 9-17.](#)
- [“Verifying the ERP Configuration” on page 9-19.](#)

ERP Specifications

ITU-T G.8032/Y.1344	Ethernet Ring Protection (Hold-off timer not supported) (Non-revertive mode not supported)
ITU-T Y.1731/IEEE 802.1ag	ERP packet compliant with OAM PDU format for CFM
Supported Platforms	OmniSwitch 10K
Maximum number of rings per node	4
Maximum number of ring ports per node	8
Maximum number of nodes per ring	16 (recommended)
Maximum number of VLANs per port.	4094
Range for ring ID	1 - 2147483647
Range for remote MEPID	1 - 8191
Range for wait-to-restore timer	1 - 12 minutes
Range for guard timer	1 - 200 centi-seconds

ERP Defaults

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

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 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.

ERP Terms

Ring Protection Link (RPL)—A designated link between two ring nodes that is blocked to prevent a loop on the ring.

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.

Link Monitoring—Ring links are monitored using standard ETH 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 (e.g., SF) on the node.

Ring APS (R-APS) 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.

ERP Protected VLAN—A VLAN that is added to the ERP ring. ERP determines the forwarding state of protected VLANs.

ERP Timers

Wait To Restore (WTR) Timer. To prevent link flapping, this timer is used by the RPL to verify that the ring has stabilized. This 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 “[ERP Specifications](#)” on page 9-2 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 “[ERP Specifications](#)” on page 9-2 for timer defaults and valid ranges.

How Does ERP Work?

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.

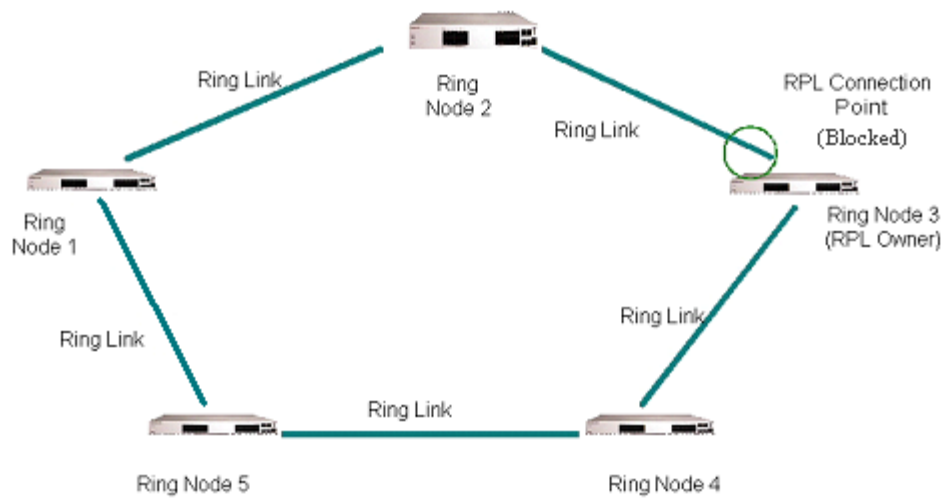
In an Ethernet ring, each node is connected to two adjacent nodes using two independent links called ring links. A ring link is bound by two adjacent nodes on ports called ring ports. The ring nodes support standard FDB (Filtering database) MAC learning, forwarding, flush behavior, and port blocking 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).

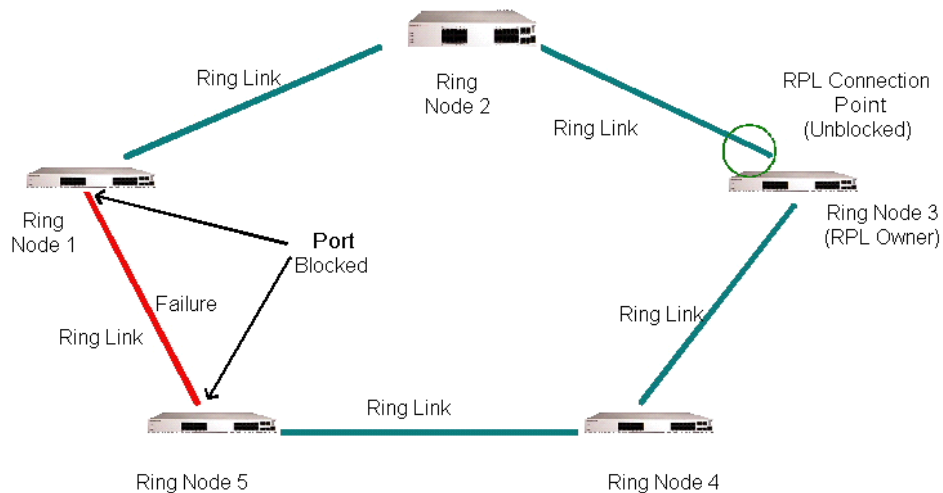
The following illustration shows an example of an ERP ring operating in the idle mode; all ring nodes are up and the RPL is blocked:



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:



Protection Mode

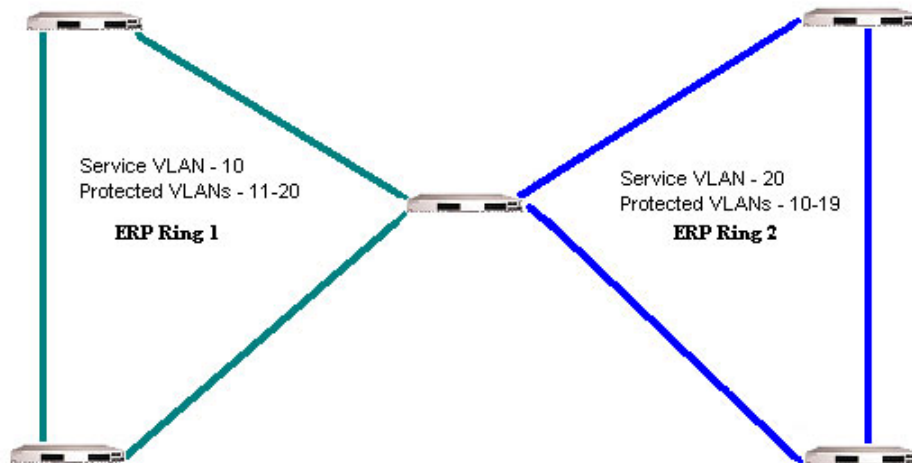
When the failed link shown in the above illustration recovers, the ring transitions as follows back to the idle mode:

- 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.



Overlapping Protected VLANs

In this example each of the ERP rings has a different Service VLAN configured which allows the ERP PDUs to be processed by the corresponding ERP ring node. The protected VLANs can be shared across ERP rings. The Service VLAN configured for ERP ring 1 is configured as a protected VLAN for ERP ring 2 and the Service VLAN for ERP ring 2 is configured as a protected VLAN for ERP ring 1.

- The two ERP rings must be configured with the same VLAN type; standard or VLAN stacking.
- Only a single node can be shared between the ERP rings; multiple shared nodes could result in a loop if the rings that are configured on shared nodes have common protected VLANs.
- Traffic for a protected VLAN is not passed if the protected VLAN is deleted from either ERP ring or if the shared node goes down.

ERP and RRSTP Differences

ERP and the Ring Rapid Spanning Tree Protocol (RRSTP) are both used for the prevention of loops in ring-based topologies but have the following differences in their implementation and functionality:

- RRSTP uses a different destination MAC address for each ring, based on the ring ID. ERP uses the same destination MAC address for all ERP protocol frames and identifies the ring based on a unique Service VLAN associated with each ring, which carries the ERP protocol frames.
- When a link failure is detected, RRSTP quickly sets the blocking ports to a forwarding state but relies on MSTP for actual protocol convergence. ERP does not require any support from MSTP. ERP has an inherent mechanism to recover from a failed state once the failed link is active again.
- MSTP determines which ports of a fully active RRSTP ring are blocked. The blocked ports (Ring Protection Link) for an ERP ring is pre-determined and configured by the user.
- RRSTP requires a ring of contiguous RRSTP nodes. ERP allows non-ERP nodes to participate in the ring by using the connectivity monitoring capabilities of Ethernet OAM to alert ERP of a link failure through non-ERP nodes.

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.

Spanning Tree

ERP has the following interactions with Spanning Tree.

- Disabling Spanning Tree on the ring ports is required before changing the switch Spanning Tree operating mode from 1x1 to flat mode.

1X1 Mode

- 1X1 STP and ERP can co-exist on different ports on the same switch but not on the same VLAN-port association (VPA). STP continues to operate as usual on non-ERP ring ports even for the ERP Protected VLANs. On the ERP ring ports, the forwarding state is controlled by ERP.

Flat Mode

- MSTP and ERP can co-exist on the same switch but are not supported on the same MSTI. ERP Protected VLANs can not be part of the same MSTI as non-ERP Protected VLANs.
- RSTP and ERP can co-exist on a node only if STP is disabled on ERP ports, the default-VLAN of ERP ports is disabled, and ERP protected VLANs are not configured on non-ERP ports. Also, non-ERP Protected VLANs must not be configured on ERP ports.
- RRSTP and ERP cannot be configured on the same port.

VLAN Stacking

The VLAN Stacking application has the following interactions with ERP:

- ERP is supported on Network Network Interface (NNI) ports; it is not supported on UNI ports.
- Tunneling of STP BPDUs across ERP links is not supported. However, tunneling of STP BPDUs across UNI ports is supported in a VLAN stacking configuration.

See [“Configuring ERP with VLAN Stacking NNIs” on page 9-14](#) for more information.

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 **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 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 **ethernet-service svlan** 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 9-9](#).

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 9-12](#).
- 2** Tag VLANs for ring protection. See [“Adding VLANs to Ring Ports” on page 9-12](#).
- 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 9-13](#).
- 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 9-13](#).
- 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 9-14](#).
- 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 9-14](#).
- 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 9-14](#).
- 8** Clear ERP statistics. Commands to clear ERP statistics for a single ring or multiple rings are described in [“Clearing ERP Statistics” on page 9-16](#).

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), VLAN Stacking User Network Interface (UNI) ports, or RRSTP ring ports.
- An ERP ring port can belong to only one ERP ring at a time.
- When configuring a ring for a switch that is operating in the flat Spanning Tree mode using STP or RSTP (not MSTP), administratively disable the default VLAN for the ring port. In this case, if the switch is using RSTP, disabling Spanning Tree on the ring port is also required.
- When configuring a ring for a switch that is operating in the flat Spanning Tree mode using MSTP, make sure the standard VLAN to which the ring port is assigned is not a member of an MSTI that is also associated with ERP protected VLANs.
- The specified service VLAN ID must not participate in a Spanning Tree instance that is associated with non-ERP VLANs. This can require changing the Spanning Tree configuration for the VLAN ID prior to using this command.

- 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. For example:

```
-> vlan 500
```

- 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 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 untagged** 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 untagged** 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 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 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 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 vlan nni** command. For example:

```
-> ethernet-service vlan 1001 nni port 1/1
-> ethernet-service vlan 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 8021q 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 via UNI ports, then STP BPDUs can be tunneled with the help of VLAN-stacking mechanism.

- Deleting an ERP service VLAN and its associated NNI ports is only allowed when the ERP ring itself is deleted using the **no** form 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 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
```

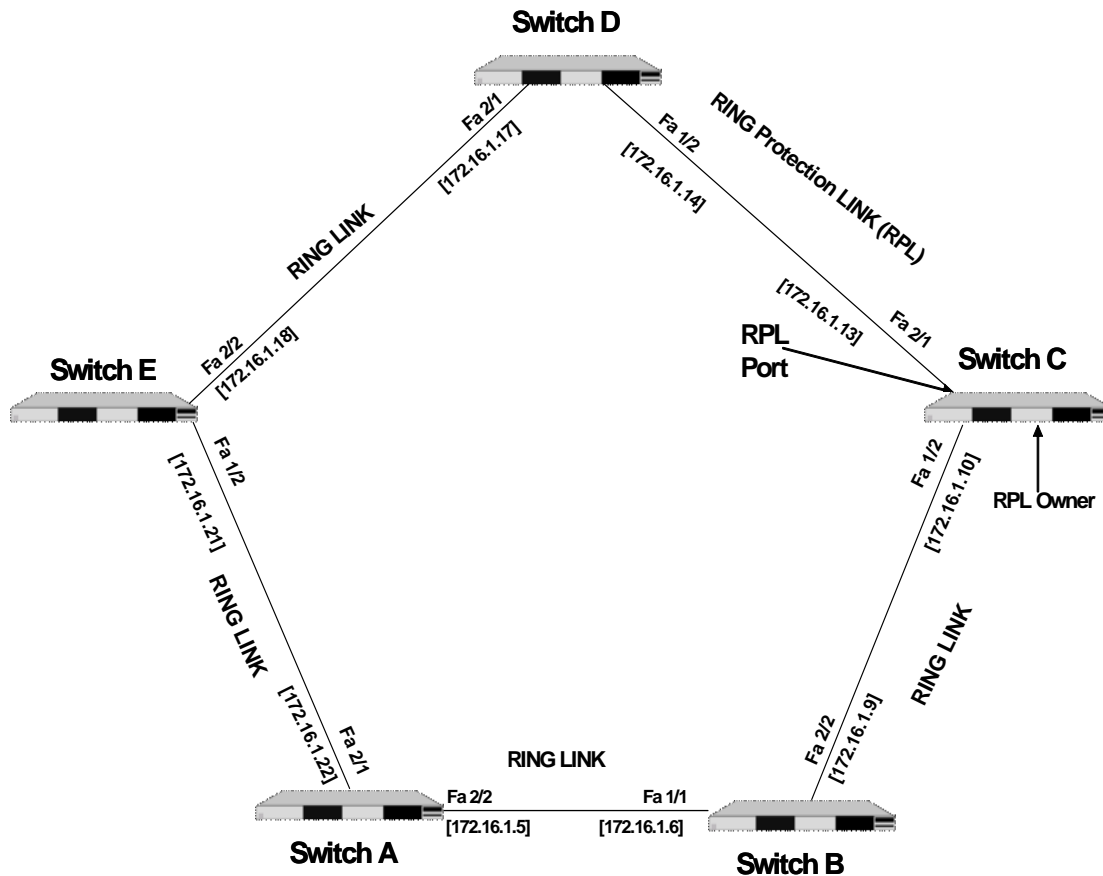
Use the **show erp statistics** command to verify ERP statistics. For more information about this command, see the *OmniSwitch CLI Reference Guide*.

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:



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 9-17](#):

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: * - Inactive Configuration

Ring Id           : 1,
Ring Port1       : 2/1,
Ring Port2       : 1/2,
Ring Status      : enabled,
Service VLAN     : 10,
WTR Timer (min)  : 5,
Guard Timer (centi-sec) : 50,
MEG Level        : 2,
Ring State       : idle,
Ring Node Type   : rpl,
RPL Port         : 2/1,
Last State Change : SUN DEC 25 06:50:17 2016 (sysUpTime 00h:01m:31s)
```

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 1/2
Legend: * - Inactive Configuration

Ring-Id : 1
Ring Port Status      : forwarding,
Ring Port Type        : non-rpl,
Ethoam Event          : disabled
```

The above command shows the forwarding status of the port, the type of ring port (RPL or non-RPL), and ETHOAM event status.

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.
show ethernet-service vlan	Displays a list of SVLANs configured from the switch.

For more information about the displays that result from these commands, see the *OmniSwitch CLI Reference Guide*.

10 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.

The LLDP-MED capability of OmniSwitch supports the usage of LLDP MED Network Policy to advertise a VLAN to the connected MEDs.

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 9, “802.1AB Commands,”](#) in the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include the following:

- [“Quick Steps for Configuring 802.1AB”](#) on page 10-2
- [“Quick Steps for Configuring LLDP-MED Network Policy”](#) on page 10-4
- [“Configuring LLDPDU Flow”](#) on page 10-10.
- [“Enabling and Disabling Notification”](#) on page 10-10.
- [“Enabling and Disabling Management TLV”](#) on page 10-10.
- [“Enabling and Disabling 802.1 TLV”](#) on page 10-11.
- [“Enabling and Disabling 802.3 TLV”](#) on page 10-11.
- [“Enabling and Disabling MED TLV”](#) on page 10-12.
- [“Setting the Transmit Interval”](#) on page 10-12.
- [“Setting the Transmit Hold Multiplier Value”](#) on page 10-12.
- [“Setting the Transmit Delay”](#) on page 10-12.
- [“Setting the Transmit Fast Start Count”](#) on page 10-13
- [“Setting the Reinit Delay”](#) on page 10-13.
- [“Setting the Notification Interval”](#) on page 10-13.

- [“Verifying 802.1AB Configuration”](#) on page 10-14.

802.1AB Specifications

Platforms Supported	OmniSwitch 10K
IEEE Specification	IEEE 802.1AB-2005 Station and Media Access Control Connectivity Discovery
Maximum number of network policies that can be associated with a port	8
Maximum number of network policies that can be configured on the switch	32

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 LLDPDU	lldp transmit interval	30 seconds
Transmit hold multiplier value	lldp transmit hold-multiplier	4
Transmit delay	lldp transmit delay	2 seconds
Reinit delay	lldp reinit delay	2 seconds
Notification interval	lldp notification interval	5 seconds
LLDPDU 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 LLDPDU 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 minimum time interval between successive LLDPDUs. To set the interval for a 20 second delay, use the **lldp transmit delay** command. For example:

```
-> lldp transmit delay 20
```

6 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           |           TLV           |           Device           |
Slot/Port| Tx      Rx      Errors  Discards | Unknown  Discards | Ageouts
-----+-----+-----+-----+-----+-----+-----+-----+-----+
  1/23   | 52      0      0      0      | 0         0         | 0
  2/47   | 50      50     0      0      | 0         0         | 0
  2/48   | 50      50     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),
  Capabilities Supported = none supported,
  Capabilities Enabled   = none enabled,
```

For more information about this display, see the *OmniSwitch CLI Reference Guide*.

Quick Steps for Configuring LLDP-MED Network Policy

LLDP-MED Network Policy for Fixed Ports

Create a VLAN, and associate a port to the VLAN. Subsequently, a network policy ID can be created and associated to the related port. The **lldp tlv med**, **lldp network-policy**, and **lldp med network-policy** commands must be used to configure and enable network policy for fixed ports.

1 Enable the transmission of network policy through a VLAN port using the **lldp tlv med** command. Configure the LLDP-MED TLVs to be transmitted through a particular port using this command. For example:

```
-> lldp 1/10 tlv med network-policy enable
```

2 Configure a local network policy on the switch for a specific application type using the **lldp network-policy** command. Assign a network policy identifier (ID) to a particular application type using this command. For example:

```
-> lldp network-policy 1 application voice vlan 10 l2-priority 5
```

3 Bind the network policy to the VLAN port using the **lldp med network-policy** command. For example:

```
-> lldp 1/10 med network-policy 1
```

4 re LLDP-MED TLVs for a particular port using this command.

```
-> lldp port 3/10 tlv med network-policy enable
```

5 Configure a local network policy on the switch for a specific application type using the **lldp network-policy application** command.

```
-> lldp network-policy 1 application voice vlan 10 l2-priority 5
```

6 Bind the network policy to a port associated with a VLAN using the **lldp med** command.

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 LLDPDU through all ports on which the protocol is enabled. If the protocol is disabled on a port, then LLDPDU received on that port are dropped.

The LLDPDU 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 through MDI TLV (in network connectivity TLV set, Extended Power-through-MDI TLV is supported)
- Link Aggregation TLV
- Maximum frame size TLV

ANSI-TIA LLDP-MED TLV Sets

- Network connectivity TLV set
- LLDP-MED capabilities TLV
- Network Policy 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). LLDP-MED Network Policy TLVs are used to let the OmniSwitch advertise the VLAN to the connected MEDs.
- Support for receiving and storing of Inventory Management TLVs from remote Media Endpoint Devices.

VLAN assignment through explicit LLDP-MED Network Policy is supported on the OmniSwitch AOS.

- The LLDP-MED service advertises the information over the Logical Link-Layer Control Frames and records higher layer management reachability and connection endpoint information from adjacent devices.
- The LLDP-MED service enabled on OmniSwitch operates in advertising mode. However, it does not support any means for soliciting information from the MEDs.

LLDP-MED Network Policy

The network policies for MED devices can be configured on the OmniSwitch using the LLDP-MED CLI commands.

Network Policy - Application Types Supported

Each network policy can be configured with one application type as a mandatory parameter. The following application types are supported:

- Voice
- Voice Signaling
- Guest Voice
- Guest Voice Signaling

- Soft phone voice
- Video Conferencing
- Streaming voice
- Video Signaling

LLDP-MED Network Policy for VLAN Advertisement

The following provisions are provided in the OmniSwitch AOS to assign LLDP-MED network policy for VLAN advertisement:

- The OmniSwitch AOS allows the configuration of a maximum of 32 network policy IDs.
- Each network policy identifier (ID) must be configured with an application type and VLAN-ID as mandatory parameters. Other parameters include L2 priority and DSCP.
- Upto 8 network policy IDs; one per each application type; can be configured for a given port.
- Two or more network policy IDs with the same application type can not be assigned to a port.
- When any MED connects to a port with an explicit MED network policy configuration, the OmniSwitch advertises the policy in the LLDPDU along with the MED Network Policy TLVs. This advertisement occurs only if the transmission of the Network Policy TLV is enabled by the user. The Media Endpoint Device must configure itself according to the advertised policy.

Fast Restart of LLDP on Detection of MED

The Fast Restart (as described in IEEE 802.1ab rev) is implemented on the OmniSwitch to transmit the related LLDP-MED Network Policy TLV as soon as a new MED endpoint is detected. The MED TLVs are encapsulated in the LLDPDU. The transmission of LLDP-MED TLV starts only when the OmniSwitch detects a MED capable endpoint on the VLAN port.

LLDP-MED for IP Phones

The LLDP-MED feature on OmniSwitch for voice transmission and VoIP Phones provides a network friendly solution. The information received from and transmitted to IP phones is tagged with voice VLAN ID.

A VLAN can be explicitly assigned to IP Phones through explicit definition of an LLDP-MED network policy identifier. The LLDP-MED Network Policy for the voice and voice signalling application must be activated on the OmniSwitch to advertise the VLAN to the connected IP Phones. For example on how to setup LLDP-MED for IP Phones, see [“Enabling and Disabling Notification” on page 10-10](#)

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.

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
```

To enable the voice application network policy for a MED TLV on the port 3 of slot 4, enter the following command at the CLI prompt:

```
-> lldp port 4/3 tlv med network policy 1 enable
```

To disable a MED TLV voice network policy on the port 3 of slot 4, enter the following command at the CLI prompt:

```
-> lldp port 4/3 tlv med network policy 1 disable
```

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 Transmit Delay

To set the minimum time interval between successive LLDPDUs transmitted, enter the **lldp transmit delay** command. For example, to set the transmit delay value to 20 seconds, enter:

```
-> lldp transmit delay 20
```

Setting the Transmit Fast Start Count

To set the fast start count in order to transmit the LLDP-MED Network Policy TLV in LLDPDU as soon as the OmniSwitch detects a new MED capable endpoint device, enter the **lldp transmit fast-start-count** command.

```
-> lldp transmit fast-start-count 3
```

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.

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 network-policy	Displays the MED Network Policy details for a given policy ID.
show lldp med network-policy	Displays the network policy configured on a slot or port. If no option is specified, network policies configured on all ports of the chassis are displayed.
show lldp remote-system	Displays local port information of remote system.
show lldp remote-system med	Displays MED local port information of remote system.

For more information about the resulting display, see [Chapter 9, “802.1AB Commands,”](#) in the *OmniSwitch CLI Reference Guide*.

11 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 15, “Configuring RIP,”](#) in this manual; or “Configuring OSPF” in the *OmniSwitch AOS Release 7 Advanced Routing Configuration Guide*.

There are two versions of Internet Protocol supported, IPv4 and IPv6. For more information about using IPv6, see [Chapter 13, “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 (e.g., `ip default-ttl`). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*. This chapter provides an overview of IP and includes information about the following procedures:

- IP Forwarding
 - Configuring an IP Router Interface (see [page 11-7](#))
 - Creating a Static Route or Recursive Static Route (see [page 11-11](#))
 - Creating a Default Route (see [page 11-12](#))
 - Configuring Address Resolution Protocol (ARP) (see [page 11-12](#))
- IP Configuration
 - Configuring the Router Primary Address (see [page 11-16](#))
 - Configuring the Router ID (see [page 11-16](#))
 - Configuring the Time-to-Live (TTL) Value (see [page 11-17](#))
 - Configuring Route Map Redistribution (see [page 11-17](#))
 - IP-Directed Broadcasts (see [page 11-23](#))
 - Protecting the Switch from Denial of Service (DoS) attacks (see [page 11-23](#))

-
- Managing IP
 - Internet Control Message Protocol (ICMP) (see [page 11-29](#))
 - Using the Ping Command (see [page 11-32](#))
 - Tracing an IP Route (see [page 11-33](#))
 - Displaying TCP Information (see [page 11-33](#))
 - Displaying User Datagram Protocol (UDP) Information (see [page 11-33](#))
 - Service Assurance Agent (SAA) (see [page 11-34](#))
 - Tunneling
 - Generic Routing Encapsulation ([page 11-34](#))
 - IP Encapsulation within IP ([page 11-34](#))
 - Tunneling operation ([page 11-35](#))
 - Configuring a Tunnel Interface ([page 11-35](#))

IP Specifications

Note that the maximum limit values provided in the following Specifications table are subject to available system resources:

Platforms Supported	OS10K
RFCs Supported	791–Internet Protocol 792–Internet Control Message Protocol 826–An Ethernet Address Resolution Protocol 2784–Generic Routing Encapsulation (GRE) 2890–Key and Sequence Number Extensions to GRE (extensions defined are not supported) 1701–Generic Routing Encapsulation (GRE) 1702–Generic Routing Encapsulation over IPV4 Networks 2003–IP Encapsulation within IP
Maximum router interfaces per system	4094 IP
Maximum router interfaces per VLAN	16
Maximum HW routes per switch	256K
Maximum SW routes per switch	N/A
Maximum static routes per switch	4K
Maximum ARP entries per module	16K
Maximum ARP entries per system	64K
Maximum number of GRE tunnel interfaces per switch	511
Maximum number of IPIP tunnel interfaces per switch	511
Routing protocols supported over the tunnel interfaces	RIP, OSPF, BGP
Maximum number of ECMP entries	512
Maximum next hops per ECMP entry	16

IP Defaults

The following table lists the defaults for IP configuration through the **ip** command.

Description	Command	Default
IP-Directed Broadcasts	ip directed-broadcast	off
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 switch, devices connected to ports on the same VLAN are able to communicate at Layer 2. The initial configuration for all Alcatel-Lucent switches 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. 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.

To forward packets to a different VLAN on a switch, you must create a router interface on each VLAN. The following steps show you how to enable IP forwarding between VLANs “from scratch”. If active VLANs have already been created on the switch, you only need to create router interfaces on each VLAN (Steps 5 and 6).

- 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** Create an IP router interface on VLAN 1 using the **ip interface** command. For example:

```
-> ip interface vlan-1 address 171.10.1.1 vlan 1
```

- 6** Create an IP router interface on VLAN 2 using the **ip interface** command. For example:

```
-> ip interface vlan-2 address 171.11.1.1 vlan 2
```

Note. See [Chapter 4, “Configuring VLANs.”](#) for more information about how to create VLANs and VLAN router interfaces.

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 several Layer 3 and Layer 4 protocols. These protocols are built into the base code loaded on the switch. A brief overview of supported IP protocols is included below.

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 simply 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. But some 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 17, “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 7 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

There are several additional IP-related protocols that 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 11-12.](#)
- Virtual Router Redundancy Protocol (VRRP)—Used to back up routers. For more information, see [Chapter 18, “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 11-29.](#)
- Multicast Services—Includes IP multicast switching (IPMS). For more information, see [Chapter 20, “Configuring IP Multicast Switching.”](#)

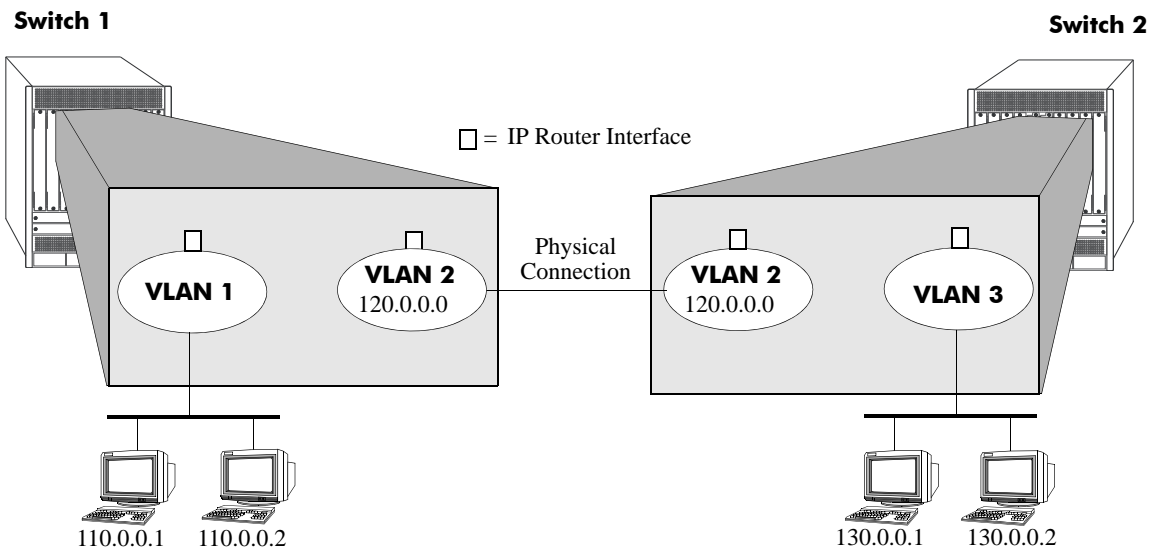
IP Forwarding

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 makes the decision on where to forward packets based on the packet's destination MAC address; routing makes the decision on where to forward packets based on the IP network address of the packet (e.g., IP - 21.0.0.10).

Alcatel-Lucent switches 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 multinetting is also supported. A network is said to be multinetted 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.



IP Forwarding

Configuring an IP Router 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, you must create at least one router 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 router interface (e.g., 193.204.173.21). Note that 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 (e.g., 255.255.0.0) or with a slash (/) after the IP address followed by the number of bits to specify the mask length (e.g., 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 11-14](#).
- 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 first 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
```

Note that when changing the IP address for the interface, the subnet mask reverts back 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, simply 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 CLI Reference Guide*.

Removing an IP Router Interface

To remove an IP router interface, use the **no** form of the **ip interface** command. Note that 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 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. This differs from other IP interfaces in that 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 the interface has to be removed using the **no ip interface Loopback0** command and re-add it 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 (i.e., 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 need to 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 7 Advanced Routing Configuration Guide* for more information.

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. You must specify the destination IP address of the route as well as the IP address of the first hop (gateway) 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, you would enter:

```
-> ip static-route 171.11.0.0 gateway 171.11.2.1
```

The subnet mask is not required if you want to use the natural subnet mask. 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, you must enter a subnet mask. For example, to create a static route to IP address 10.255.11.0, you would have to 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
```

Note that 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; you must delete them from the table. Use the **no ip static route** command to delete a static route. You must 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 171.11.0.0 gateway 171.11.2.1
```

The IP 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 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. The path to a recursive route can be changed dynamically if a better route to the gateway is learned. 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 171.11.0.0 follows 192.168.10.1
```

A route to the **192.168.10.1** address would need to 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. You must 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
```

Note that 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 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 **vrf** command to add a permanent entry to the ARP table. You must 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, you would enter:

```
-> 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
```

Note that 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. Note that 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.** Use the **arp-name** parameter to specify a name for the permanent ARP entry.

For example:

```
-> arp 171.11.1.1 00:2a:90:d1:8e:10 arp-name server1
```

Use the **show arp** command to display the ARP table.

Note. Because most hosts support the use of address resolution protocols to determine and cache address information (called dynamic address resolution), you generally do not need 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, you would enter:

```
-> 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. Permanent entries must be cleared 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 via the router. ARP packets are still bridged across multiple ports.

Note that Local Proxy ARP takes precedence over any switch-wide Proxy ARP or ARP function. In addition, it is not necessary to configure Proxy ARP in order 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
```

Note that 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 this 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.

ARP Filtering

ARP filtering is used to determine whether or not the switch responds to ARP requests that contain a specific IP address. This feature is generally used in conjunction with the Local Proxy ARP application; however, ARP filtering is available for use on its own and/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 (e.g., 193.204.173.21) used to determine whether or not an ARP packet is filtered.
- An IP mask (e.g. 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 about this and other ARP filter commands, see the *OmniSwitch CLI Reference Guide*.

IP Configuration

IP is enabled on the switch by default and there are few options that can, or need to be, configured. This section provides instructions for some 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. The router primary IP address is used by BGP to derive its unique BGP Identifier, if the router `router-id` is not a valid IP unicast address.

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 would 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
```

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 would 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

It is possible to learn and advertise IPv4 routes between different protocols. Such a process is referred to as route redistribution and is configured using the **ip redistrib** 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, it is given a name to identify the group of statements that it represents. This name is required by the **ip redistrib** command. Therefore, configuring route redistribution involves the following steps:

- 1 Create a route map, as described in [“Using Route Maps” on page 11-17](#).
- 2 Configure redistribution to apply a route map, as described in [“Configuring Route Map Redistribution” on page 11-21](#).

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	ip-address	metric
deny	ip-nexthop	metric-type
	ipv6-address	tag
	ipv6-nexthop	community
	tag	local-preference
	ipv4-interface	level
	ipv6-interface	ip-nexthop
	metric	ipv6-nexthop
	route-type	

Refer to the “IP Commands” chapter in the *OmniSwitch 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 redistrib** command. See “[Configuring Route Map Redistribution](#)” on page 11-21 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-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 redistrib** 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-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. Because 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 `redstripv4`:

```
-> no ip route-map redstripv4
```

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 `redstripv4` route map:

```
-> no ip route-map redstripv4 sequence-number 10
```

Note that in the above example, the `redstripv4` 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 `redstripv4` sequence 10:

```
-> no ip route-map redstripv4 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 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 can 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 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 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 redistrib-control all-
subnets
-> ipv6 access-list ip6addr address 2001::1/64 action permit redistrib-control no-
subnets
```

For more information about configuring access list commands, see the “IP Commands” chapter in the *OmniSwitch CLI Reference Guide*.

Configuring Route Map Redistribution

The **ip redistrib** 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 perform 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 a BGP network using the `ospf-to-bgp` route map:

```
-> ip redistrib 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 11-17 for more information.

To remove a route map redistribution configuration, use the **no** form of the **ip redistrib** command. For example:

```
-> no ip redistrib ospf into bgp route-map ospf-to-bgp
```

Use the **show ip redistrib** command to verify the redistribution configuration:

```
-> show ip redistrib
```

Source Protocol	Destination 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 redistrib** command. For example, the following command disables the redistribution administrative status for the specified route map:

```
-> ip redistrib ospf into bgp route-map ospf-to-bgp status disable
```

The following command example enables the administrative status:

```
-> ip redistrib 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 redistrib 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 (those not processed by sequence 10 or 20) and sets the tag for such routes to eight.

IP-Directed Broadcasts

An IP directed broadcast is an IP datagram that has all zeroes 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 “smurf” attacks. In a smurf 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. Typically, directed broadcasts must not be enabled.

Use the `ip directed-broadcast` command to enable or disable IP-directed broadcasts. For example:

```
-> ip directed-broadcast off
```

Use the `show ip config` command to display the IP-directed broadcast state.

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 of these attacks aim at system bugs or vulnerability (for example, teardrop attacks), while other types of attacks involve generating large volumes of traffic so that network service is denied to legitimate network users (such as peps attacks). 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 hang or crash the system.
- **SYN Attack**—Floods a system with a series of TCP SYN packets, resulting in the host issuing SYN-ACK responses. The half open TCP connections can exhaust TCP resources, such that no other TCP connections are accepted.
- **Land Attack**—Spoofed packets are sent with the SYN flag set to a host on any open port that is listening. The machine can hang or reboot in an attempt to respond.
- **Pepsi Attack**—The most common form of UDP flooding directed at harming networks. A pepsi attack is an attack consisting of a large number of spoofed UDP packets aimed at diagnostic ports on network devices. This can cause network devices to use up a large amount of CPU time responding to these packets.
- **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. Examples of some invalid source and destination IP addresses are listed below:

Invalid Source IP address	<ul style="list-style-type: none">• 0.x.x.x.• 255.255.255.255.• subnet broadcast, i.e. 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	<ul style="list-style-type: none"> • 127.x.x.x. • in the range 240.x.x.x - 255.255.255.254. • 0.0.0.0 (valid exceptions - certain DHCP packets e.g.). • 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.

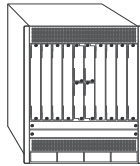
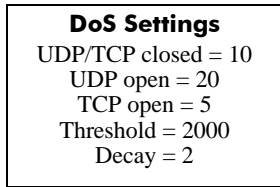
Note. In both the conditions described above in “Multicast IP and MAC Address Mismatch”, packets are dropped and SNMP traps are generated.

- 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 because 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 (loopback network) are received by the switch. When such packets are detected, they are dropped, and SNMP traps are generated.

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:

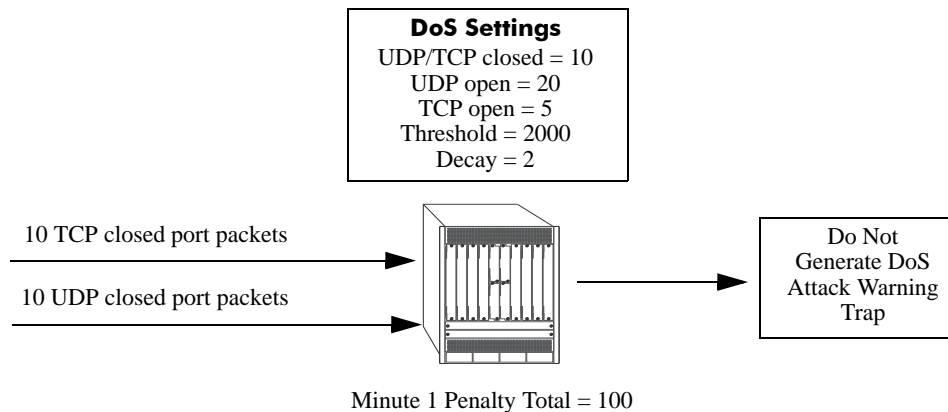


Penalty Total = 0

In one minute, 10 TCP closed port packets and 10 UDP closed port packets are received. This would bring the total penalty value to 200, as shown using the following equation:

$$(10 \text{ TCP} \times 10 \text{ penalty}) + (10 \text{ UDP} \times 10 \text{ penalty}) = 200$$

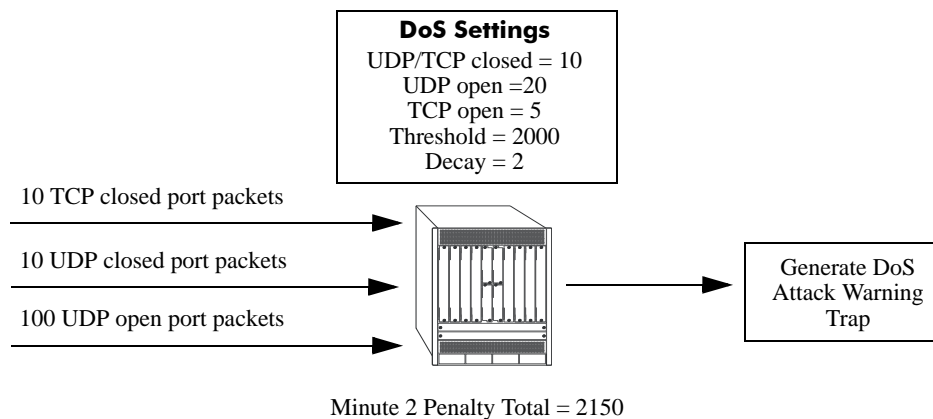
This value would be divided by 2 (due to the decay) and decreased to 100. The switch would not record a port scan:



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:

$$(100 \text{ previous minute value}) + (10 \text{ TCP} \times 10 \text{ penalty}) + (10 \text{ UDP} \times 10 \text{ penalty}) + (200 \text{ UDP} \times 20 \text{ penalty}) = 4300$$

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:



The above functions and how to set their values are covered in the sections that follow.

Setting Penalty Values

There are three types of traffic you can set a penalty value for:

- 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

DoS traps must be enabled in order 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.

Alcatel-Lucent 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 **arp filter** 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 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, etc., 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 selectively disable (close) TCP/UDP well-known service ports and enable them when necessary. This command only operates on TCP/UDP ports that are opened by default. It has no effect on ports that are opened by loading applications, such as RIP and BGP.

In addition, the **ip service** command allows you to designate which port to enable or disable by specifying the name of a service or the well-known port number associated with that service. For example, both of the following commands disable the telnet service:

```
-> no ip service telnet
-> no ip service port 23
```

Note that specifying a port number requires the use of the optional **port** keyword.

To enable or disable more than one service in a single command line, enter each service name separated by a space. For example, the following command enables the telnet, ftp, and snmp service ports:

```
-> ip service telnet ftp snmp
```

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
network-time	123
snmp	161

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 several 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. This prevents 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 might be unreachable. 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, some devices might continue using the less-efficient route.
- **Time-Exceeded Message**—Sent by the switch if an IP packet's TTL field reaches zero. The TTL field prevents packets from continuously circulating the internetwork if the internetwork contains a routing loop. Once a packet's 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, several commonly used ICMP messages have been separate CLI commands for convenience. These commands are listed below with 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 10, "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 icmp control** 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 destination's IP address or 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 **tracroute** 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 some timing information. When using this command, you must 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) that will 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.

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. But some 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 Alcatel-Lucent implementation provides support for two tunneling protocols: Generic Routing Encapsulation (GRE) and IP encapsulation within IP (IPIP).

Generic Routing Encapsulation

GRE encapsulates a packet that needs 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 payload's IP header.

Consider the following when configuring the GRE tunnel interfaces:

- A switch can support up to 127 GRE tunnel interfaces.

IP Encapsulation within IP

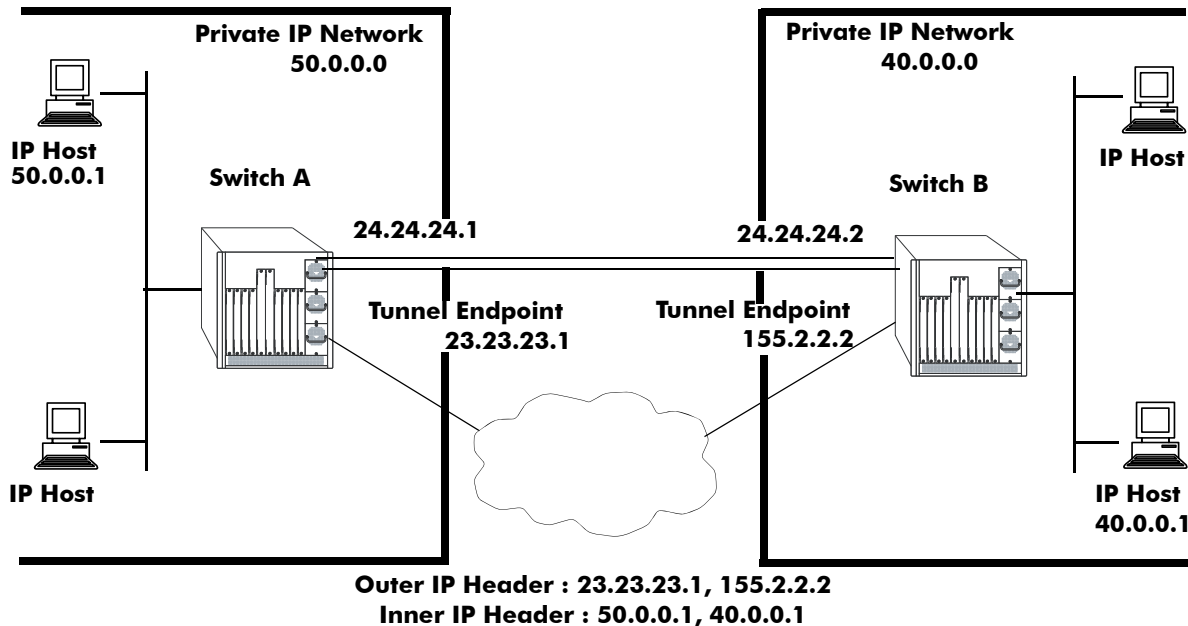
IPIP tunneling is a method by which an IP packet is encapsulated within another IP packet. The Source Address and Destination Address of the outer IP header identifies the endpoints of tunnel. Whereas Source Address and Destination Address of the inner IP header identifies the original sender and recipient 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 diagram below illustrates how packets are forwarded over the tunnel.



In the above 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
```

Note. An interface can be configured only as a VLAN or a Tunnel interface.

Note. To display information about the configured tunnels on the switch, use the **show ip interface**.

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 current status of TCP/UDP service ports. Includes service name and well-known port number.
show arp	Displays the ARP table.
show arp filter	Displays the ARP filter configuration for the switch.
show icmp control	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 CLI Reference Guide*.

12 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 CLI Reference Guide*. This chapter provides an overview of Multiple VRF and includes the following information:

- [Quick Steps for Configuring Multiple VRF](#).
- [Using the VRF Command Line Interface](#).
- [VRF Interaction With Other Features](#).
- [Configuring VRF Instances](#).
- [Verifying the VRF Configuration](#)

VRF Specifications

Routing Protocols Supported	Static, IPv4, RIPv2, OSPFv2,BGP4
Maximum VRF instances per switch	64
Maximum VRF instances per VLAN	1
Maximum OSPFv2 VRF routing instances per switch	16
Maximum RIPv2 VRF routing instances per switch	16
Maximum BGP VRF routing instances per switch	32
SNMP version required for management	SNMPv3

VRF Defaults

Parameter Description	Command	Default Value/Comments
Active VRF instance	vrf	Default VRF instance

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. In addition, if the name specified does not exist, a VRF instance is automatically created. 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. For example:

```
-> vrf IpOne
IpOne: ->
```

Note that in the above 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 IpTwo
IpTwo: ->
```

Note. In the above 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
```

Note. Verify the Multiple VRF configuration using the **show vrf** command:

```
IpOne: -> show vrf
Virtual Routers      Protocols
-----
      default
      IpOne      RIP
      IpTwo      BGP

Total Number of Virtual Routers: 3
```

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 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 IpOne
IpOne: -> vrf 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
```

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.

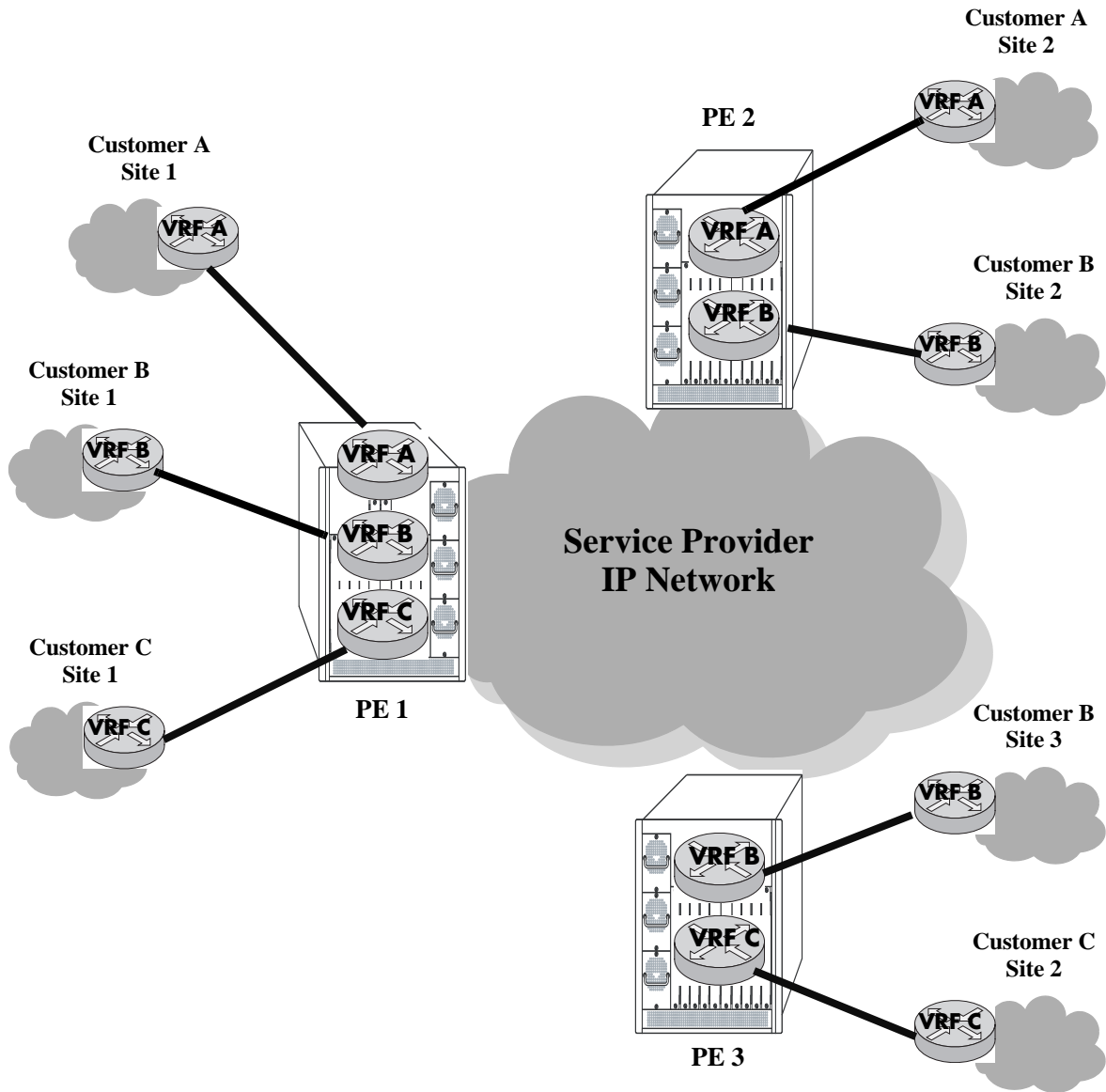
Some of the benefits of using the Multiple VRF feature include the following:

- 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 12-7](#) 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.



Example Multiple VRF Configuration

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 12-9](#)

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 12-12](#) and [“Verifying the VRF Configuration” on page 12-15](#).

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 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.

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	
BFD	IPv6 (NDP/Tunnel)	DNS Client
Static routes	BGPv6	Telnet Client
IPv4/ARP	RIPng	FTP Client/Server
RIPv2	IS-IS	SSH Client
BGPv4	OSPFv3	AAA
OSPFv2	DVMRP	NTP
PIM-DM (IPv4)	SFTP	Trap Manager
PIM-SM (IPv4)	Policy Based Routing	SNMP (Agent)
Route Map Redistribution	Router Discovery Protocol	HTTP Server
IP-IP Tunnels	EMP access	Webview
GRE Tunnels		
Ping		
Traceroute		
SSH Server (SSH, SFTP, SCP)		
Telnet Server		
VRRPv2/VRRPv3		
QoS VRF Policies		
UDP/DHCP Relay		
AAA RADIUS Server		

The following subsections provide additional information related to Multiple VRF interaction with specific applications.

AAA RADIUS Servers

- AAA RADIUS servers can be configured on any VRF instance including the default VRF instance. However, all of the RADIUS servers must reside on the same VRF instance.
- The VRF instance that the RADIUS 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 RADIUS servers reside on is deleted or disabled, access to the RADIUS servers is disabled as well.

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.

Management Applications (Telnet and SSH)

- Telnet and SSH (ssh, sftp, and scp) sessions “to” the switch are now VRF aware. Client support for these utilities is supported only in the default VRF instance.
- A maximum of 4 combined Telnet sessions are allowed simultaneously across all VRFs on the switch.
- A maximum of 8 combined SSH sessions are allowed simultaneously across all VRFs on the switch

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:

```
-> qos policy condition c1 vrf engr_vrf
-> qos policy condition c2 vrf default
-> qos 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 12-14](#) 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 creating a VRF instance, assigning one or more IP interfaces to the instance, and configuring routing protocols to operate within a specific instance.

The initial configuration of an Alcatel-Lucent switch 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 IpOne
IpOne: ->
```

In this example, instance *IpOne* is created and made the active VRF context at the same time. Note that 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.

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.

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.

Note. If the instance name specified with the **vrf** command does not exist, a VRF instance is automatically created. In addition, 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.

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 12-9](#). 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 12-3](#).

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 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      Protocols
-----
      default
      IpOne      RIP
      IpTwo      BGP

Total Number of Virtual Routers: 3
```

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 CLI Reference Guide*.

13 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 CLI Reference Guide*.

This chapter provides an overview of IPv6 and includes information about the following procedures:

- Configuring an IPv6 interface (see [page 13-13](#))
- Configuring a Unique Local Ipv6 Interface (see [page 13-13](#))
- Assigning IPv6 Addresses (see [page 13-15](#))
- Configuring IPv6 Tunnel Interfaces (see [page 13-17](#))
- Creating a Static Route (see [page 13-18](#))
- Configuring the Route Preference of a Router (see [page 13-19](#))
- Configuring Route Map Redistribution (see [page 13-20](#))

IPv6 Specifications

Note that the maximum limit values provided in the following Specifications table are subject to available system resources:

Platforms Supported	OS10K
RFCs Supported	2460–Internet Protocol, Version 6 (IPv6) Specification 2461–Neighbor Discovery for IP Version 6 (IPv6) 2462–IPv6 Stateless Address Autoconfiguration 2464–Transmission of IPv6 Packets Over Ethernet Networks 3056–Connection of IPv6 Domains via IPv4 Clouds 4213–Basic Transition Mechanisms for IPv6 Hosts and Routers 4291–Internet Protocol Version 6 (IPv6) Addressing Architecture 4443–Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification 1493 - Unique Local IPv6 Unicast Address
Maximum IPv6 interfaces	VLANs- 4096 Configured Tunnels - 255 6to4 Tunnels - 1
Maximum IPv6 global unicast addresses	4K
Maximum IPv6 global unicast addresses per IPv6 interface	50
Maximum IPv6 routes when there are no IPv4 routes present (includes dynamic and static routes)	OS10K-GNI-C48/U48 - 8K OS10K-XNI-U32 - 6K (Note: Exceeding these limits, or having IPv4 routes will result in some traffic being routed in software)
Maximum IPv6 static routes per switch	4K
Maximum Number of RIPng Peers	10
Maximum Number of RIPng Interfaces	10
Maximum Number of RIPng Routes	512

IPv6 Defaults

The following table lists the defaults for IPv6 configuration through the **ip** 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

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 13-6](#) 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 [“Auto-configuration of IPv6 Addresses” on page 13-8](#) 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 bits 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.

Note. IPv6 does not support the use of broadcast addresses. This functionality is replaced using improved multicast addressing capabilities.

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 was 1234:531F: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 bits 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 or anycast 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 13-15](#) 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 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 13-17](#) 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.

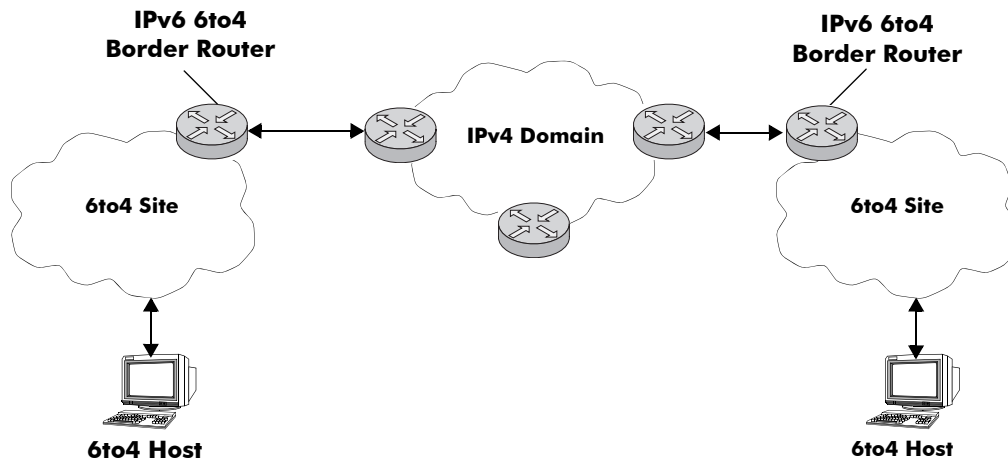
An IPv6 6to4 tunnel interface is identified by its assigned address, which is derived by combining a 6to4 well-known prefix (2002) with a globally unique IPv4 address and embedded as the first 48 bits of an IPv6 address. For example, 2002:c633:6489::137/16, where D467:8A89 is the hex equivalent of the IPv4 address 198.51.100.137.

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:



In the above diagram:

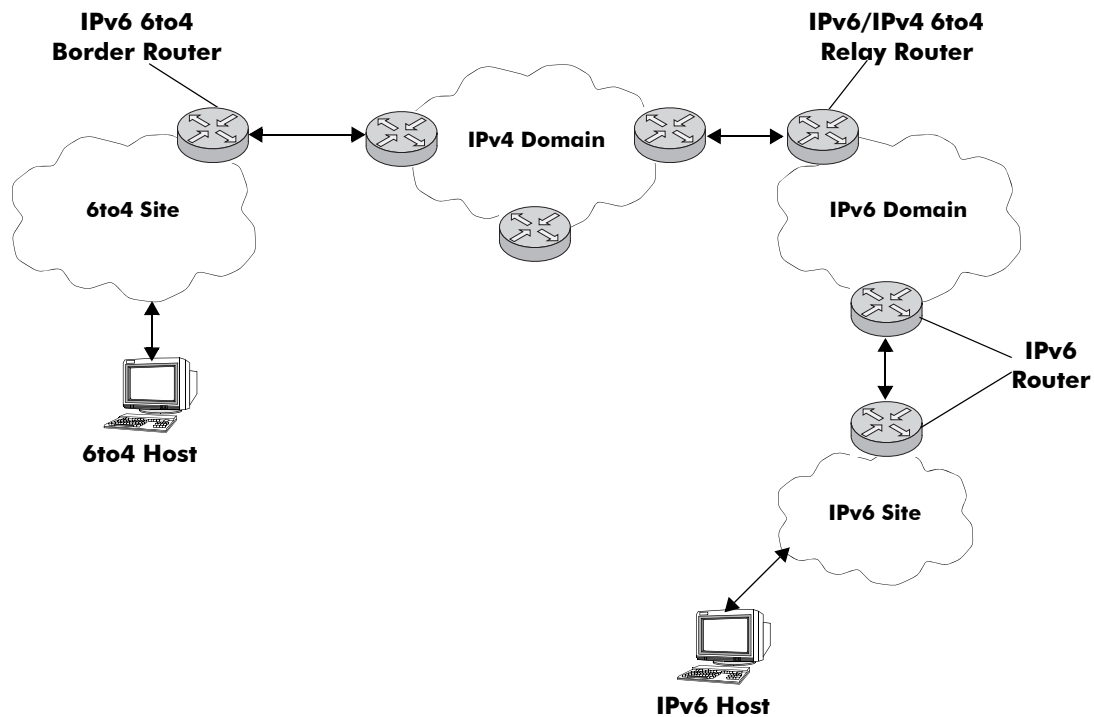
- 1 The 6to4 hosts receive 6to4 prefix from Router Advertisement.
- 2 The 6to4 host sends IPv6 packets to 6to4 border router.
- 3 The 6to4 border router encapsulates IPv6 packets with IPv4 headers and sends to the destination 6to4 border router over the IPv4 domain.
- 4 The destination 6to4 border router 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:



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 13-13](#) and [“Configuring IPv6 Tunnel Interfaces” on page 13-17](#). 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 13-17](#). For more detailed information about configured tunnels, refer to RFC 4213.

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 13-17](#) 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 13-8](#).
- 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.
- Each IPv6 interface anycast address must also have a unique prefix. However, multiple devices may share the same anycast address prefix to identify themselves as members of the anycast group.

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 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
```


Assigning IPv6 Addresses

As was previously mentioned, 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 (e.g., global unicast or anycast) 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 13-6](#) 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 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 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.

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.

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
  RIP           120
  EBGp          190
  IBGP          200
```

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 redistrib** 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 redistrib** command. Therefore, configuring route redistribution involves the following steps:

- 1 Create a route map, as described in [“Using Route Maps” on page 13-20](#).
- 2 Configure redistribution to apply a route map, as described in [“Configuring Route Map Redistribution” on page 13-24](#).

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 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 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 redistrib** command. See [“Configuring Route Map Redistribution” on page 13-24](#) 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 redistrib** 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 `redistipv4` 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 redistrib-control all-
subnets
-> ipv6 access-list ip6addr address 2001::1/64 action permit redistrib-control no-
subnets
```

For more information about configuring access list commands, see the “IP Commands” chapter in the *OmniSwitch CLI Reference Guide*.

Configuring Route Map Redistribution

The **ipv6 redistrib** 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 redistrib 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 13-20](#) for more information.

To remove a route map redistribution configuration, use the **no** form of the **ipv6 redistrib** command. For example:

```
-> no ipv6 redistrib ospf into rip route-map ospf-to-rip
```

Use the **show ipv6 redistrib** command to verify the redistribution configuration:

```
-> show ipv6 redistrib
```

Source Protocol	Destination 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 redistrib** command. For example, the following command disables the redistribution administrative status for the specified route map:

```
-> ipv6 redistrib ospf into rip route-map ospf-to-rip admin-state disable
```

The following command example enables the administrative status:

```
-> ipv6 redistrib 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 redistrib 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.

Verifying the IPv6 Configuration

A summary of the show commands used for verifying the IPv6 configuration is given here:

show ipv6 rip	Displays the RIPng status and general configuration parameters.
show ipv6 redistrib	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 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.

For more information about the displays that result from these commands, see the *OmniSwitch CLI Reference Guide*.

14 Configuring IPsec

Internet Protocol security (IPsec) is a suite of protocols for securing IPv6 communications by authenticating and/or encrypting each IPv6 packet in a data stream. IPsec is a framework of open standards designed to provide interoperable, high quality, cryptographically-based security for IPv6 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.

Note. The OmniSwitch currently supports IPsec for IPv6 only.

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 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Master Key Configuration (see [“Configuring an IPsec Master Key”](#) on page 14-9).
- Security Policy Configuration (see [“Configuring an IPsec Policy”](#) on page 14-10).
- Security Policy Rule Configuration (see [“Configuring an IPsec Rule”](#) on page 14-13).
- SA Configuration (see [“Configuring an IPsec SA”](#) on page 14-14).
- Security Association Key Configuration (see [“Configuring IPsec SA Keys”](#) on page 14-15).
- Discard Policy Configuration (see [“Assigning an Action to a Policy”](#) on page 14-12)

IPsec Specifications

Platforms Supported	OmniSwitch 10K
IP Version Supported	IPv6
RFCs Supported	4301 - Security Architecture for the Internet Protocol 4302 - IP Authentication Header (AH) 4303 - IP Encapsulating Security Payload (ESP) 4305 - Cryptographic Algorithm Implementation Requirements for ESP and AH 4308 - Cryptographic Suites for IPsec
Encryption Algorithms Supported for ESP	NULL, 3DES-CBC, and AES-CBC
Key lengths supported for Encryption Algorithms	3DES-CBC - 192 bits AES-CBC - 128, 192, or 256 bits
Authentication Algorithms Supported for AH	HMAC-SHA1-96, HMAC-MD5-96, and AES-XCBC-MAC-96
Key lengths supported for Authentication Algorithms	HMAC-MD5 - 128 bits HMAC-SHA1 - 160 bits AES-XCBC-MAC - 128 bits
Master Security Key formats	Hexadecimal (16 bytes) or String (16 characters)
Priority value range for IPsec Policy	1 - 1000
Index value range for IPsec Policy Rule	1 - 10
SPI Range	256 - 999999999
Modes Supported	Transport

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)		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 policies 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 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
```

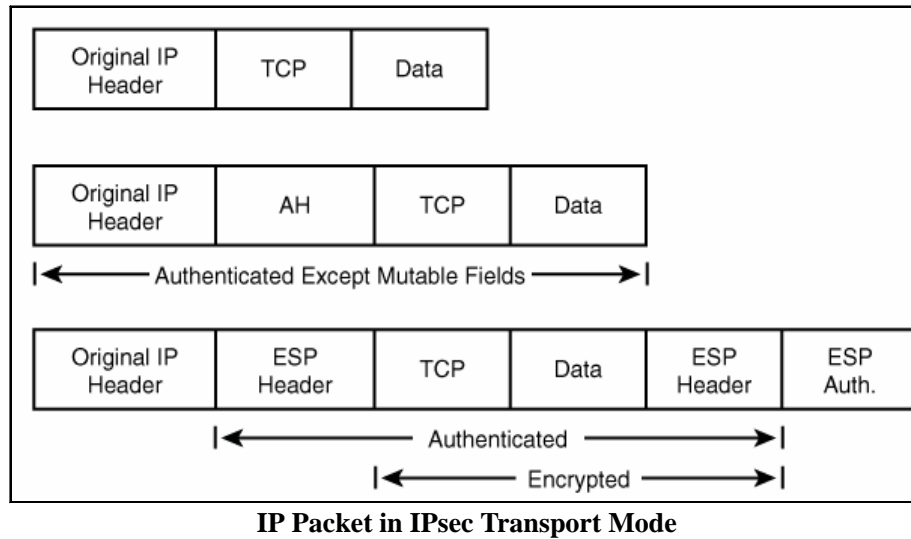
```
-> show ipsec ipv6 statistics
```

IPsec Overview

IPsec provides protection to IPv6 traffic. To achieve this, IPsec provides security services for IPv6 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 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 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 packet is encapsulated, and an IPsec header (AH or ESP) is inserted between the original IPv6 header and the upper-layer protocol header. The figure below shows an IPv6 packet protected by IPsec in transport mode.

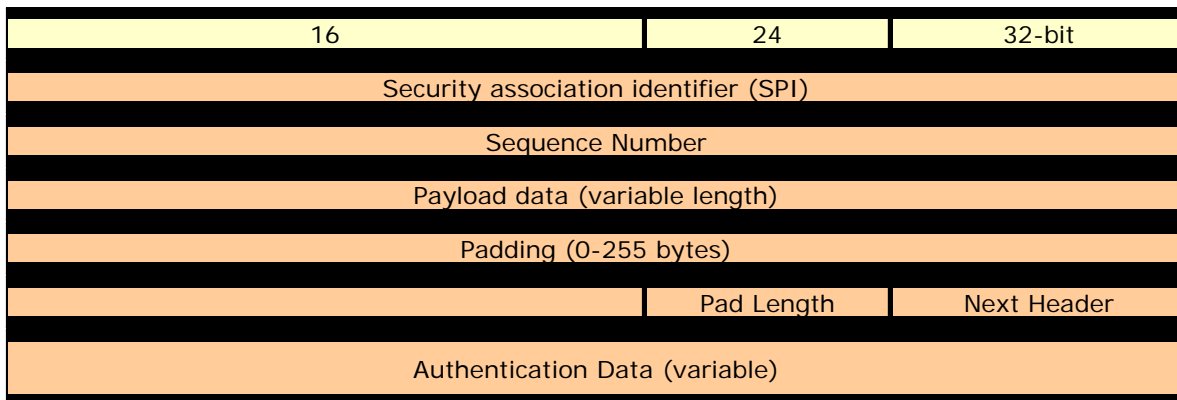


Note. The OmniSwitch currently supports the Transport Mode of operation.

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.



IP Packet protected by ESP

ESP is identified by a value of 50 in the IPv6 header. The ESP header is inserted after the IPv6 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 IPv6 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 IPv6 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 plaintext 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.

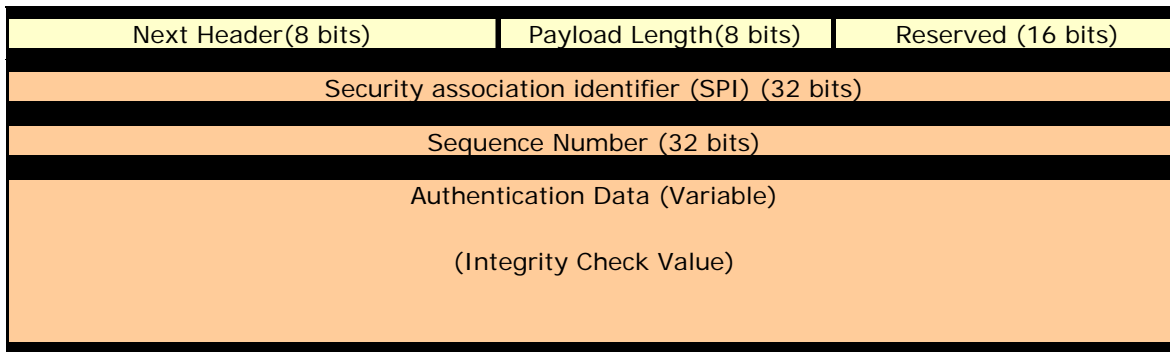
Authentication Header (AH)

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.



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 OmniSwitch

IPsec allows the following 3 types of actions to be performed on an IPv6 datagram that matches the filters defined in the security policy:

- The IPv6 datagram can be subjected to IPsec processing, i.e. encrypted, and/or authenticated via ESP and AH protocols.
- The IPv6 datagram can be discarded.
- The IPv6 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 specify 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 config 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 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 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 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.”](#)

Configuring IPsec for securing IPv6 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 14-9.](#)
- Configure an IPsec Security Policy on the switch. This is described in [“Configuring an IPsec Policy” on page 14-10.](#)
- Set an IPsec rule for the configured IPsec Security Policy on the switch. This is described in [“Configuring an IPsec Rule” on page 14-13.](#)
- Enable the Security Policy. This is described in [“Enabling and Disabling a Policy” on page 14-11.](#)
- Configure the authentication and encryption keys required for manually configured IPsec Security associations (SA). This is described in [“Configuring IPsec SA Keys” on page 14-15](#)
- 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 14-14.](#)

Configuring IPsec for discarding IPv6 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 packets. This is described in [“Discarding Traffic using IPsec” on page 14-8.](#)

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.

Note. 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).

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 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 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 IPv6 address, destination IPv6 address and optional parameters such as IPv6 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 proto-
col 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 14-13](#).

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 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 IPv6 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
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.

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 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 IPv6 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, IPv6 source address, IPv6 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-sha1 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.

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
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.

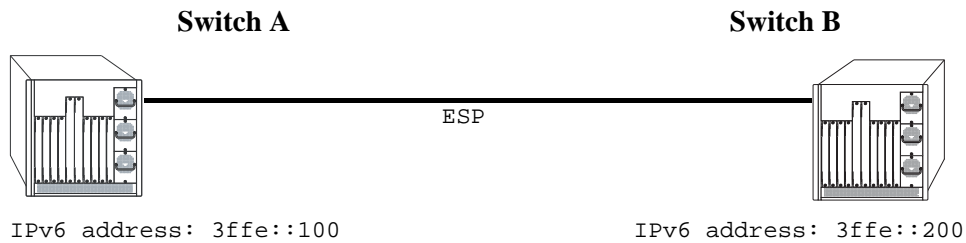
Once IPsec is configured for IPv6 on the switch, you can monitor the incoming and outgoing packets for the configured parameters by using the **show ipsec ipv6 statistics** command.

```
Inbound:
  Successful                = 2787
  Policy violation          = 0
  No SA found               = 0
  Unknown SPI               = 0
  AH replay check failed   = 0
  ESP replay check failed  = 0
  AH authentication success = 93
  AH authentication failure = 0
  ESP authentication success = 25
  ESP authentication failure = 0
  Packet not valid         = 0
  No memory available       = 0
Outbound:
  Successful                = 5135
  Policy violation          = 0
  No SA found               = 19
  Packet not valid         = 0
  No memory available       = 0
```

Additional Examples

Configuring ESP

The example below shows the commands for configuring ESP between two OmniSwitches for all TCP traffic.



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-shal 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-shal description "ESP from 100" admin-
state enable

-> ipsec key tcp_out_esp sa-encryption 12345678

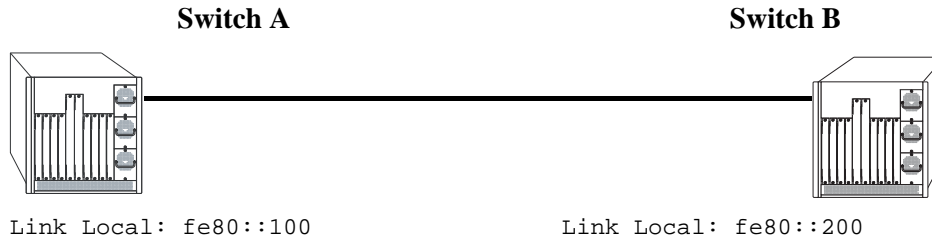
-> ipsec key tcp_out_esp sa-authentication 12345678901234567890

-> 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.



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 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 ipv6 statistics	Displays IPsec statistics for IPv6 traffic.

For more information about the resulting displays from these commands, see the “IPsec Commands” chapter in the *OmniSwitch CLI Reference Guide*.

Examples of the above commands and their outputs are given in the section “[Configuring IPsec on the OmniSwitch](#)” on page 14-9

15 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 CLI Reference Guide*.

This chapter provides an overview of RIP and includes information about the following procedures:

- RIP Routing
 - Loading RIP (see [page 15-6](#))
 - Enabling RIP (see [page 15-7](#))
 - Creating a RIP Interface (see [page 15-7](#))
 - Enabling a RIP Interface (see [page 15-7](#))
- RIP Options
 - Configuring the RIP Forced Hold-Down Interval (see [page 15-9](#))
 - Configuring the RIP Update Interval (see [page 15-9](#))
 - Configuring the RIP Invalid Timer (see [page 15-10](#))
 - Configuring the RIP Garbage Timer (see [page 15-10](#))
 - Configuring the RIP Hold-Down Timer (see [page 15-10](#))
 - Enabling a RIP Host Route (see [page 15-11](#))
- RIP Redistribution
 - Configuring Route Redistribution (see [page 15-12](#))
- RIP Security
 - Configuring Authentication Type (see [page 15-18](#))
 - Configuring Passwords (see [page 15-18](#))

RIP Specifications

Platforms Supported	OS10K
RFCs Supported	RFC 1058–RIP v1 RFC 2453–RIP v2 RFC 1722–RIP v2 Protocol Applicability Statement RFC 1724–RIP v2 MIB Extension
Maximum Number of Interfaces	10
Maximum Number of Peers	100
Maximum Number of Routes	10K
Maximum Number of ECMP Entries	512
Maximum Number of ECMP Next Hops	16

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 Alcatel-Lucent's optional Advanced Routing Software. 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 15-6](#).
- **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 Alcatel-Lucent's optional Advanced Routing Software. For more information see the “Configuring OSPF” chapter in the *OmniSwitch AOS Release 7 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.

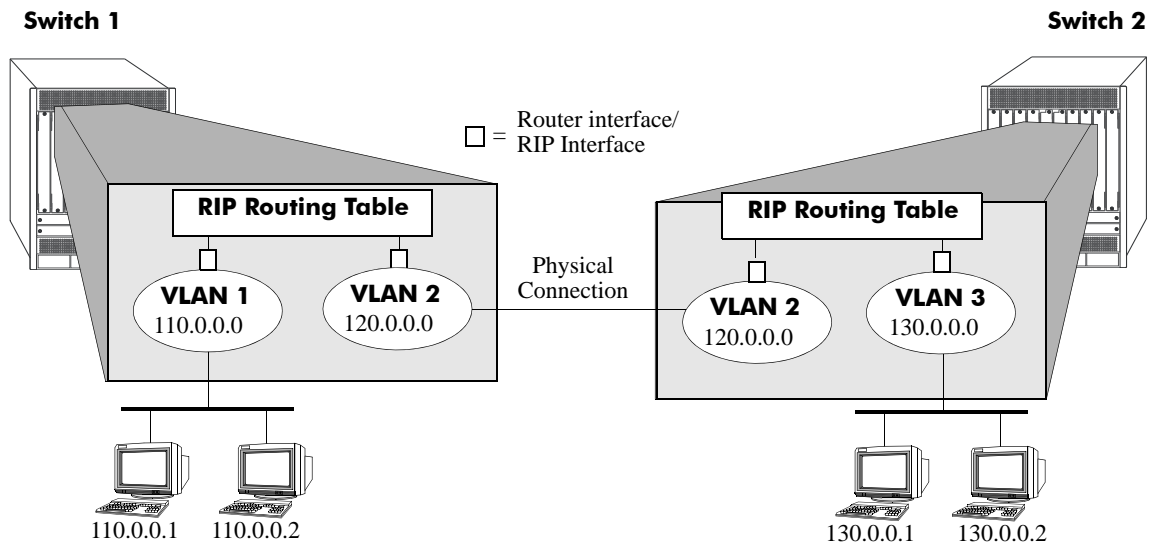
RIP Version 2

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 15-18](#).
- **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 20, “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.



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 11, "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 than 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 15-10](#).

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 hold-down 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 redistrib** 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 redistrib** command. Therefore, configuring route redistribution involves the following steps:

- 1 Create a route map, as described in [“Using Route Maps” on page 15-12](#).
- 2 Configure redistribution to apply a route map, as described in [“Configuring Route Map Redistribution” on page 15-16](#).

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 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 redistrib** command. See [“Configuring Route Map Redistribution” on page 15-16](#) 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 redistrib** 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 `redistipv4` 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 CLI Reference Guide*.

Configuring Route Map Redistribution

The **ip redistrib** 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 redistrib 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 15-12](#) for more information.

To remove a route map redistribution configuration, use the **no** form of the **ip redistrib** command. For example:

```
-> no ip redistrib ospf into rip route-map ospf-to-rip
```

Use the **show ip redistrib** command to verify the redistribution configuration:

```
-> show ip redistrib
```

Source Protocol	Destination 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 redistrib** command. For example, the following command disables the redistribution administrative status for the specified route map:

```
-> ip redistrib ospf into rip route-map ospf-to-rip admin-state disable
```

The following command example enables the administrative status:

```
-> ip redistrib 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 redistrib	Displays the currently configured RIP redistribution filters.

For more information about the displays that result from these commands, see the *OmniSwitch CLI Reference Guide*.



16 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 liveness 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 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Global Configuration (see [page 16-11](#)).
- Interface Level Configuration (see [page 16-11](#)).
- OSPF level configuration (see [page 16-15](#)).
- BGP Level Configuration (see [page 16-18](#)).
- VRRP Level Configuration (see [page 16-19](#)).
- Static Routing Level Configuration (see [page 16-21](#)).

BFD Specifications

Platforms Supported	OmniSwitch 10K
RFCs Supported	RFC 5880 - Bidirectional Forwarding Detection RFC 5881 - Bidirectional Forwarding Detection for IPv4 and IPv6 (Single Hop) RFC 5882 - Generic Application of Bidirectional Forwarding Detection
Protocols Supported	BGP, OSPF, VRRP Remote Address Tracking only, and Static Routes. IPv6 protocols not supported.
Modes Supported	Asynchronous Echo (Demand Mode not supported)

BFD Defaults

The following table shows the default settings of the configurable 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
Administrative status of a BFD interface	ip bfd interface admin-state	Disabled
Transmit time interval for a BFD interface.	ip bfd interface transmit	300 milliseconds
Receive time interval for the BFD interface.	ip bfd interface receive	300 milliseconds
BFD interface detection time multiplier.	ip bfd interface multiplier	3
Echo time interval for the BFD interface	ip bfd interface echo-interval	300 milliseconds
BFD status for the OSPF protocol	ip ospf bfd-state	Disabled
BFD status for an OSPF interface	ip ospf interface bfd-state	Disabled
BFD session status with all BGP neighbors	ip bgp bfd-state all-neighbors	Disabled

Parameter Description	Command	Default Value/Comments
BFD session status with all neighbors of the corresponding interface which are greater than or equal to “2-way” state	ip ospf interface bfd-state all-neighbors	Enabled
BFD status for the BGP protocol	ip bgp bfd-state	Disabled
BFD status for BGP neighbors	ip bgp neighbor bfd-state	Disabled
BFD status for VRRP protocol	vrrp bfd-state	Disabled
BFD status for a VRRP tracking policy.	vrrp track address bfd-state	Enabled
BFD status for a static route.	ip static-route bfd-state	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 16-5).

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 IP interface using the **ip bfd interface** command. For example:

```
-> ip bfd interface
```

Optional: Configure the BFD session explicitly with an IP interface name if non-default BFD session parameters are required for BFD sessions that must be run separate from the IP interface.

```
-> ip bfd interface bfd_int_1
```

- 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 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
```

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 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
```

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 BFD session detection time multiplier value using the **ip bfd interface multiplier** command. For example:

```
-> ip bfd interface bfd-vlan-101 multiplier 5
```

Note. Demand mode is not supported. The default operational mode is Asynchronous with the echo function enabled. However, Static Routing and VRRP protocol support BFD in the echo-only operational mode.

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 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
```

10 Optional: Enable the administrative status of a BFD interface using the **ip bfd interface admin-state** command. For example:

```
-> ip bfd interface bfd-vlan-101 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.* Verify the BFD interface session status and configuration using the [show ip bfd interfaces](#) command. For example:

```
-> show ip bfd interfaces one
Interface Name           = one,
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
```

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
```

See the “BFD Commands” chapter in the *OmniSwitch 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:

- BGP
- OSPF
- VRRP Tracking
- Static routes

Once the BFD configuration is in place (see “[Quick Steps for Configuring BFD](#)” on page 16-3), the steps described in the following sections are used to configure BFD interaction with the supported Layer 3 protocols.

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 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 BGP

1 Register BGP with the BFD protocol using the **ip bgp bfd-state** command. For example:

```
-> ip bgp bfd-state enable
```

2 Enable BFD for specific BGP neighbors using the **ip 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
-> ip bgp bfd-state all-neighbors enable
```

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 address bfd-state** command. For example:

```
-> vrrp track 2 address 10.1.1.1 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 ID	Policy	Admin State	Oper State	Pri	BFD Status
1	25.25.25.1	Enabled	Down	50	Enabled
2	192.10.150.42	Enabled	Down	25	Enabled

See the “VRRP Commands” chapter in the *OmniSwitch CLI Reference Guide* for information about the fields in this display.

Configuring BFD Support for Static Routes

Enable BFD support for a specific static route using the **ip static-route bfd-state** command or for all 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 a static route, make sure that:

- the gateway address does not match any of the local interface addresses on the switch
- BFD is enabled on 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 static routes on which BFD is enabled use the [show ip router database](#) command along with the **protocol static** option as shown below:

Example:

```
r4c1-> ip static-route 100.0.0.0/8 gateway 100.1.1.10 bfd-state enable
```

```
r4c1-> show ip router database protocol static
```

Legend: + indicates routes in-use
 b indicates BFD-enabled static route
 r indicates recursive static route, with following address in brackets

Total IPRM IPv4 routes: 7

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 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, BFP, 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 $[(\text{Detect Time Multiplier}) * (\text{Required Minimum Rx Interval})]$, the BFD session remains up. Any routing protocol that associates the BFD maintains its adjacencies and continues its periodic transmission of BFD 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 the Asynchronous mode. In this mode, 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.

In addition to the operational mode, an Echo function is available to verify the forwarding path between neighboring BFD systems. When enabled, a BFD system transmits Echo packets 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.

The Echo function is a configurable option and can work on its own or simultaneously with the Asynchronous mode. Note that using the Echo function with the Asynchronous mode lowers the rate at which control packets are sent because Echo packets are then used to detect session liveliness. In addition, transmitting Echo packets is only allowed over a single hop; transmitting BFD control packets is allowed over multiple hops.

Once a BFD session is started, the BFD peers can decide whether or not Echo packets are actually transmitted. A session is considered down when the peers receive no BFD control packets from each other or if sufficient Echo packets are missed within a specific period of time.

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 16-11](#) for more information.
- 2** Configure BFD support for the Layer 3 protocols for which BFD establishes sessions. This implementation of BFD supports the IPv4 versions of BGP, OSPF, VRRP remote tracking, and static routes. See [“Configuring BFD Support for Layer 3 Protocols” on page 16-15](#) 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 16-23](#) 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 16-12](#)).
- Transmit time interval (see [“Configuring the BFD Transmit Time interval” on page 16-12](#)).
- Receive time interval (see [“Configuring the BFD Receive Time Interval” on page 16-12](#)).
- Multiplier (see [“Configuring the BFD Multiplier” on page 16-13](#)).
- Echo interval (see [“Configuring the BFD Echo interval” on page 16-13](#)).

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 16-14](#) for more information.

Configuring a BFD Session

To configure a BFD session, use the **ip bfd interface** command and specify an existing IP interface name. For example:

```
-> ip bfd interface bfd-vlan-101
```

The above command configures BFD with the name `bfd-vlan-101`. See [“Enabling or Disabling BFD Status” on page 16-14](#) for more information.

To delete the BFD session, use the **no** form of the above command. For example:

```
-> no ip bfd interface bfd-vlan-101
```

The above command deletes the BFD session on `bfd-vlan-101`.

Note. The BFD interface session must be associated to an existing IP interface that is configured with an IP address.

Configuring the BFD Transmit Time Interval

Transmit 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 msec.

To change the transmit time interval for a specific BFD interface session, use the **ip bfd interface transmit** command along with the name and transmit time interval in milliseconds. For example:

```
-> ip bfd interface bfd-vlan-101 transmit 500
```

The above command changes the transmit time interval value to 500 msec on `bfd-vlan-101`.

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 msec.

To change the receive time interval for BFD control packets, use the **ip bfd interface receive** command. For example:

```
-> ip bfd interface bfd-vlan-101 receive 500
```

The above command changes the receive time interval value to 500 msec on `bfd-vlan-101`.

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 bfd interface echo-interval** command. For example:

```
-> ip bfd interface bfd-vlan-101 echo-interval 500
```

The above command configures the echo time interval value to 500 milliseconds on `bfd-vlan-101`.

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 bfd interface multiplier** command. For example:

```
-> ip bfd interface bfd-vlan-101 multiplier 5
```

The above command assigns a multiplier value of 5 to `bfd-vlan-101`.

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 bfd interface admin-state` command. For example:

```
-> ip bfd interface bfd-vlan-101 admin-state enable
```

The above command enables the administrative status of `bfd-vlan-101`.

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 status` command with the `disable` keyword. For example:

```
-> ip bfd interface bfd-vlan-101 admin-state disable
```

To verify the 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 msec, receive interval of 300 msec, multiplier 3, and echo interval of 300 msec.

To verify the BFD status and configuration, use the `show ip bfd interfaces` command. For example:

```
-> show ip bfd interfaces
```

Interface Name	Admin Status	Tx Interval	Min Rx Interval	Min EchoRx Interval	Detect Multiplier	OperStatus
one	enabled	300	300	300	3	UP
two	enabled	300	300	300	3	UP

The output above displays the interfaces participating in the BFD sessions, along with their IP interface names and respective BFD session parameters. To see additional detail for a specific interface, use the `show ip bfd interfaces` command and specify an interface name. For example:

```
-> show ip bfd interfaces one
Interface Name          = one,
```

```

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

```

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, OSPF, 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, OSPF, VRRP Tracking, and Static Routes:

[“Configuring BFD Support for OSPF” on page 16-15.](#)

[“Configuring BFD Support for BGP” on page 16-18.](#)

[“Configuring BFD Support for VRRP Tracking” on page 16-19.](#)

[“Configuring BFD Support for Static Routes” on page 16-21.](#)

Configuring BFD Support for OSPF

The steps below show how to configure and verify BFD support for OSPF, so that OSPF is a registered protocol with BFD and receives forwarding path detection failure messages from BFD.

Note. OSPF must be running on all participating routers, and BFD must be configured and enabled on the participating OSPF interfaces. See [“Configuring BFD Session Parameters” on page 16-11](#) for more information.

1 To associate BFD with the OSPF protocol and to change the default BFD status for the OSPF protocol, register OSPF with BFD at the protocol level using the `ip ospf bfd-state` command. For example:

```
-> ip ospf bfd-state enable
```

The BFD status for the OSPF protocol is now enabled, which means that communication between OSPF and BFD is enabled. To de-register OSPF with BFD, enter the following command:

```
-> ip ospf bfd-state disable
```

2 To verify the BFD status for OSPF protocol, use the `show ip 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

```

3 Once OSPF is registered with BFD at the protocol level, enable the OSPF interface(s) that participate in BFD using the **ip ospf interface bfd-state** command. For example:

```
-> ip ospf interface vlan-10 bfd-state enable
```

The above command enables BFD on the interface named vlan-10. To enable BFD on all configured OSPF interfaces, use the **ip ospf bfd-state all-interfaces** command. For example:

```
-> ip ospf bfd-state all-interfaces enable
```

To disable BFD for all configured OSPF interfaces, use the **ip ospf bfd-state all-interfaces** command with the **disable** keyword. For example:

```
-> ip ospf bfd-state all-interfaces disable
```

4 To display the BFD status on an OSPF interface, use the **show ip ospf interface** command. For example:

```
-> show ip ospf interface
```

Interface Name	DR Address	Backup Address	DR Status	Admin Status	Oper State	BFD 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

5 Once OSPF is registered with BFD at the protocol level and BFD is enabled on the desired OSPF interface(s), use the **show ip bfd interfaces** command to display BFD-enabled interfaces. For example:

```
-> show ip bfd interfaces
```

Interface Name	Admin Status	Tx Interval	Min Rx Interval	Min EchoRx Interval	Detect Multiplier	OperStatus
one	enabled	300	300	300	3	UP
two	enabled	300	300	300	3	UP

6 To establish BFD sessions with neighbors that are in full state only, enter the **ip ospf interface bfd-state drs-only** command as shown below:

```
-> ip ospf interface int1 bfd-state drs-only
```

The above command establishes a BFD session on interface named int1 with OSPF 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** command as shown below:

```
-> ip ospf interface int2 bfd-state all-neighbors enable
```

The above command establishes a BFD session on interface named int2 with all OSPF neighbors that are greater than or equal to “2-way” state.

When any neighbors are added to this interface, OSPF informs BFD about the newly added neighbor(s); BFD then establishes a session with them. Use the **show ip 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 Discr	Interface Name	Neighbor Address	State	Remote Discr	Neg. Rx Intvl	Neg. Tx Intvl	EchoRx Intvl
1	v1001	101.1.1.11	UP	1	300	300	300
2	v2000	200.1.1.1	UP	0	0	0	300

To view a BFD session with a particular neighbor, use the **show ip 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
```

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 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 16-11 for more information.

1 To associate BGP protocol with BFD liveness detection, register BGP with BFD at the protocol level using the **ip bgp bfd-state** 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
```

2 Once BGP is registered with BFD at the protocol level, you need to enable BFD for particular BGP neighbors using the **ip bgp neighbor bfd-state** command as shown below:

```
-> ip bgp neighbor 135.10.10.2 bfd-state enable
```

The above command enables BFD for neighbor with IP address 135.10.10.2. To enable BFD for all BGP neighbors, 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 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

Use the **show ip 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 Discr	Interface Name	Neighbor Address	State	Remote Discr	Neg. Rx Intvl	Neg. Tx Intvl	EchoRx Intvl
1	v1001	101.1.1.11	UP	1	300	300	300
2	v2000	200.1.1.1	UP	0	0	0	300

```
-> show ip bfd sessions 1
```

```
Local discriminator           = 1,
Neighbor IP Address          = 100.1.1.10,
Requested Session Type       = ASYNC ,
Interface IfIndex            = 2,
Source UDP Port              = 49152,
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:    = BGP
```

Configuring BFD Support for VRRP 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.

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: 1 second
VRRP default priority: 100
VRRP default preempt: Yes
VRRP trap generation: Enabled
VRRP startup delay: 45 (expired)
```

```
VRRP BFD-STATUS : Enabled
```

VRID	VLAN	IP Address(es)	Admin Status	Priority	Preempt	Adv. Interval
1	1	192.168.170.1 192.168.170.2	Enabled	255	Yes	1
2	15	10.2.25.254	Disabled	100	No	1

2 Once VRRP is registered with BFD at the protocol level, enable BFD for a particular VRRP track policy using the **vrrp track address bfd-state** command. Ensure that the track policy is associated with at least one of the virtual routers. For example:

```
-> vrrp track 2 address 10.1.1.1 bfd-state enable
```

The above command enables BFD for a track policy with VRRP track number 2 and a remote interface address of 10.1.1.1.

Note. The value of the address parameter should be a remote interface address. BFD cannot be configured for a local interface address.

Use the **show vrrp track** command to verify whether BFD is enabled for a particular track policy. For example:

```
-> show vrrp track
```

Track ID	Policy	Admin State	Oper State	Pri	BFD Status
1	190.1.1.10	Enabled	Down	100	Disabled
2	10.1.1.1	Enabled	Up	25	Enabled

Use the **show ip 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 IP address. BFD session is also established with the BFD neighbors.

Use the **show ip 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 Discr	Interface Name	Neighbor Address	State	Remote Discr	Neg. Rx Intvl	Neg. Tx Intvl	EchoRx Intvl
1	v1001	101.1.1.11	UP	1	300	300	300
2	v2000	200.1.1.1	UP	0	0	0	300

```
-> show ip bfd sessions 2
```

```
Local discriminator           = 2,
Neighbor IP Address          = 10.1.1.1,
Requested Session Type       = ECHO,
Interface IfIndex            = 4,
Source UDP Port               = 49153,
State                         = UP,
Session Operating Mode       = None,
Remote discriminator         = 0,
Negotiated Tx interval       = 0,
Negotiated Rx interval       = 0,
Echo Rx interval              = 300,
Multiplier                    = 3,
Applications Registered:     = VRRP
```

Configuring BFD Support for Static Routes

This section provides information about how to configure and verify BFD support for 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** command. For example :

```
-> ip static-route 10.1.1.1 mask 255.0.0.0 gateway 10.1.1.25 bfd-state enable
```

Note. Static Routes support BFD in the echo-only operational mode.

The above command enables BFD support for a static route with destination ip address as 10.1.1.1, destination network mask as 255.0.0.0, and gateway address as 10.1.1.25.

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. You can verify the BFD session list which shows the gateway address using the **show ip bfd sessions** command.

To enable BFD support for all static routes, use the **ip static-route all bfd-state** command:

```
-> ip static-route all bfd-state enable
```

To verify the static routes on which BFD is enabled, use the **show ip 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
r indicates recursive static route, with following address in brackets

Total IPRM IPv4 routes: 7

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

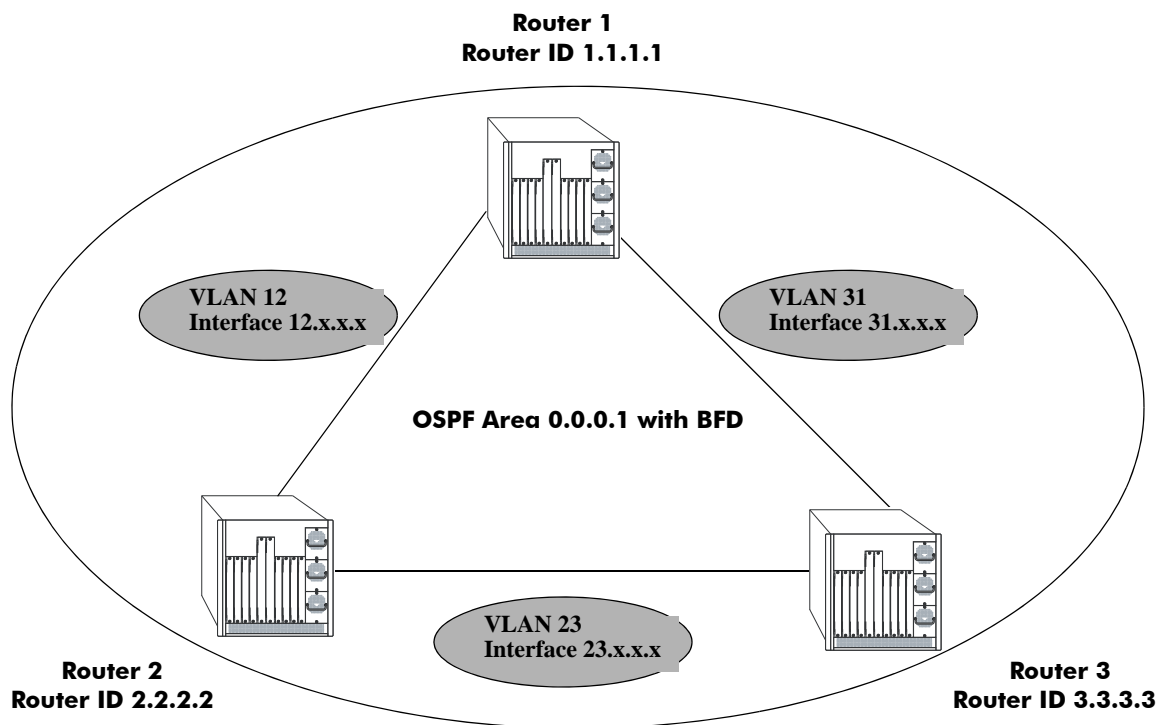
Destination	Gateway	Metric
-------------	---------	--------

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.



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 isstatically 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 16-24](#).

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 16-24](#).

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 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 bfd interfaces	Displays the BFD interface configuration for the switch.
show ip bfd sessions	Displays the BFD neighbors and session states.
show ip ospf	Displays the BFD status for the OSPF protocol.
show ip ospf interface	Displays the BFD status for OSPF interfaces.
show ip bgp	Displays the BFD status for the BGP protocol.
show ip bgp neighbors	Displays the BFD status for BGP neighbors.
show vrrp	Displays the BFD status for the VRRP protocol.
show vrrp track	Displays the BFD status for a track policy.
show ip router database protocol static	Displays the BFD status for static routes.

For more information about the resulting displays from these commands, see the *Omniswitch CLI Reference Guide*. Examples of the above commands and their outputs are given in the section “[Configuring BFD](#)” on page 16-11.

17 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 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Quick steps for configuring DHCP Relay on [page 17-4](#).
- Setting the IP address for Global DHCP on [page 17-9](#).
- Identifying the VLAN for Per-VLAN DHCP on [page 17-9](#).
- Enabling DHCP Relay on [page 17-10](#).
- Setting the Forward Delay time on [page 17-10](#).
- Setting the Maximum Hops value on [page 17-11](#).
- Setting the Relay Forwarding Option to Standard and Per-VLAN on [page 17-11](#).
- Using automatic IP configuration to obtain an IP address for the switch on [page 17-12](#).
- Configuring relay for generic UDP service ports on [page 17-13](#).

For information about the IP protocol, see [Chapter 11, “Configuring IP.”](#)

DHCP Relay Specifications

Platforms Supported	OmniSwitch 10K
RFCs Supported	0951–Bootstrap Protocol 1534–Interoperation between DHCP and BOOTP 1541–Dynamic Host Configuration Protocol 1542–Clarifications and Extensions for the Bootstrap Protocol 2132–DHCP Options and BOOTP Vendor Extensions 3046–DHCP Relay Agent Information Option, 2001
DHCP Relay Implementation	Global DHCP Per-VLAN DHCP
DHCP Relay Service	BOOTP/DHCP (Bootstrap Protocol/Dynamic Host Configuration Protocol)
UDP Port Numbers	67 for Request 68 for Response
IP addresses supported for each Relay Service	Maximum of 256 IP addresses for each Relay Service.
IP addresses supported for the Per-VLAN service	Maximum of 256 VLAN relay services.
Maximum number of UDP relay services allowed per switch	10
Maximum number of VLANs to which forwarded UDP service port traffic is allowed	256

DHCP Relay Defaults

The following table describes the default values of the DHCP Relay parameters:

Parameter Description	Command	Default Value/Comments
Default UDP service	ip udp relay service	BOOTP/DHCP
Forward delay time value for DHCP Relay	ip helper forward-delay	3 seconds
Maximum number of hops	ip helper maximum-hops	4 hops
Packet forwarding option	ip helper standard ip helper per-vlan-only	Standard
Automatic switch IP configuration for default VLAN 1	ip helper boot-up	Disabled
Automatic switch IP configuration packet type (BootP or DHCP)	ip helper boot-up enable	BootP
Relay Agent Information Option	ip helper agent-informa- tion	Disabled

Quick Steps for Setting Up DHCP Relay

You must configure DHCP Relay on switches where packets are routed between IP networks.

There is no separate command for enabling or disabling the relay service. DHCP Relay is automatically enabled on the switch whenever a DHCP server IP address is defined. To set up DHCP Relay, proceed as follows:

1 Identify the IP address of the DHCP server. Where the DHCP server has IP address 128.100.16.1, use the following command:

```
-> ip helper address 128.100.16.1
```

2 Set the forward delay timer for the DHCP relay. To set the timer for a 15 second delay, use the following command:

```
-> ip helper forward-delay 15
```

3 Set the maximum hop count value. To set a hop count of 3, use the following command:

```
-> ip helper maximum-hops 3
```

Note. Optional. To verify the DHCP Relay configuration, enter the **show ip helper** command. The display shown for the DHCP Relay configured in the above Quick Steps is shown here:

```
-> show ip helper
```

```
Ip helper :
Forward Delay (seconds) = 15,
Max number of hops      = 3,
Relay Agent Information = Disabled,
PXE support              = Disabled,
Forward option           = standard mode,
Bootup Option            = Disable
Forwarding address list (Standard mode): 128.100.16.1
```

For more information about this display, see the “DHCP Relay” chapter in the *OmniSwitch 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 standard and per-VLAN-only mode. The standard mode forwards all DHCP packets on a global relay service. The per-VLAN-only mode forwards DHCP packets that originate from a specific VLAN. See [“Setting the Relay Forwarding Option” on page 17-11](#) 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 Automatic IP Configuration” on page 17-12](#) 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 helper** and related commands.

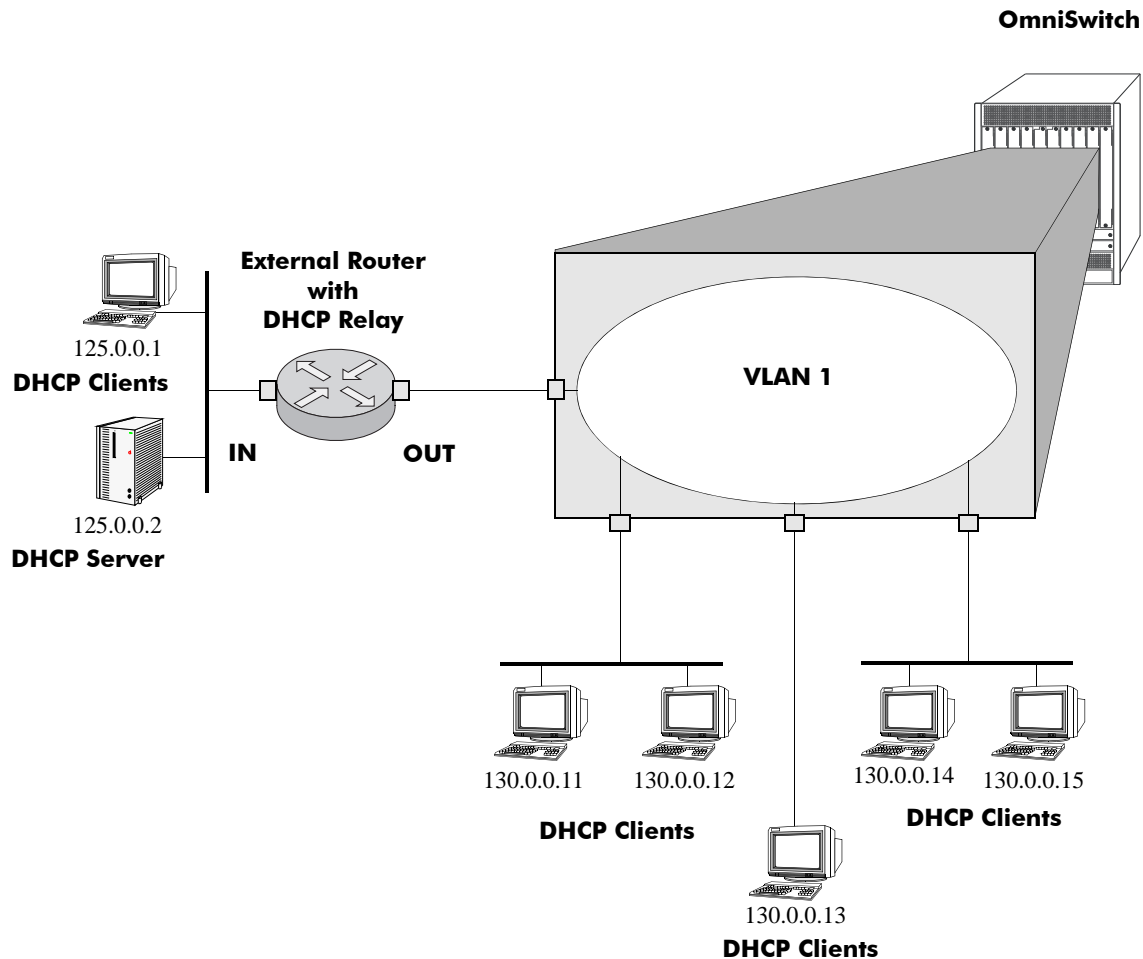
The Generic UDP Relay forwards the broadcast packets with pre-configured destination UDP port information to destination VLAN or VLANs. This feature can be configured using the **ip udp relay** and related commands.

For more information on the CLI commands related to DHCP Relay, see the *DHCP Relay Commands* chapter in the *OmniSwitch 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.



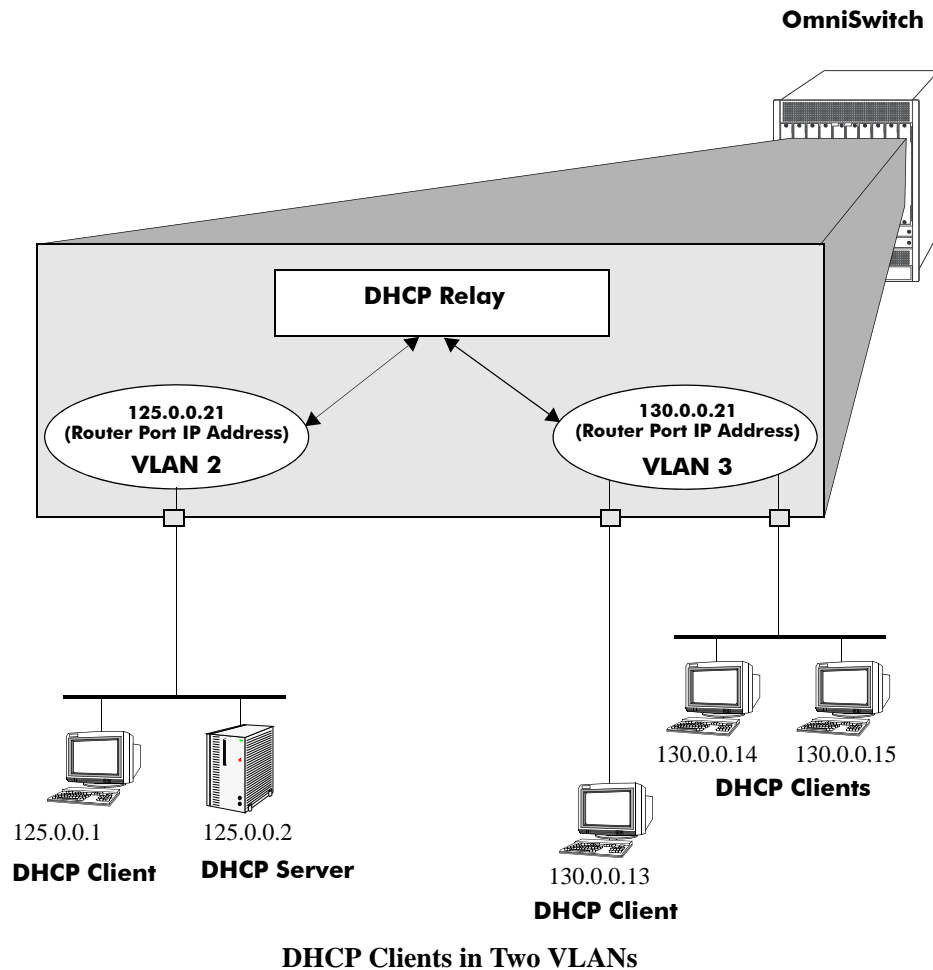
DHCP Clients are Members of the Same VLAN

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 helper address** command. For more information, see “[DHCP Relay Implementation](#)” on page 17-9.

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.



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.

DHCP Relay Implementation

The OmniSwitch allows you to configure the DHCP Relay feature in one of two ways. You can set up a global DHCP Relay or you can set up the DHCP Relay based on the DHCP packet from the client. Both of these choices provide the same configuration options and capabilities. However, they are mutually exclusive. The following matrix summarizes the options.

Per-VLAN 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	Disabled	DHCP Request is relayed to the Per-VLAN Relay
Enabled	Enabled	N/A

Global DHCP

For the global DHCP service, you must identify an IP address for the DHCP server.

Setting the IP Address

The DHCP Relay is enabled on a switch whenever a DHCP server IP address is defined by using the **ip helper address** command. There is no separate command for enabling or disabling the relay service. You must configure DHCP Relay on switches where packets are routed between IP networks. The following command defines a DHCP server address:

```
-> ip helper address 125.255.17.11
```

The DHCP Relay 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 an IP address, use the **no** form of the **ip helper address** command. The IP address specified with this syntax is deleted. If an IP address is not specified with this syntax, then *all* IP helper addresses are deleted. The following command deletes an IP helper address:

```
-> no ip helper address 125.255.17.11
```

Per-VLAN DHCP

For the Per-VLAN DHCP service, you must identify the number of the VLAN that makes the relay request.

Identifying the VLAN

You can enter one or more server IP addresses to which packets are sent from a specified VLAN. Do this by using the **ip helper vlan address** command. The following syntax identifies the IP address 125.255.17.11 as the DHCP server for VLAN 3:

```
-> ip helper address 125.255.17.11 vlan 3
```

The following syntax identifies two DHCP servers for VLAN 4 at two different IP addresses:

```
-> ip helper address 125.255.17.11 125.255.18.11 vlan 4
```

To delete an IP address, use the **no** form of the **ip helper address** command. The IP address specified with this syntax is deleted. If an IP address is not specified with this syntax, then *all* IP helper addresses are deleted. The following command deletes a helper address for IP address 125.255.17.11:

```
-> no ip helper address 125.255.17.11
```

The following command deletes all IP helper addresses:

```
-> no ip helper address
```

Configuring BOOTP/DHCP Relay Parameters

Once the IP address of the DHCP server(s) is defined and the DHCP Relay is configured for either Global DHCP request or Per-VLAN DHCP request, you can set the following optional parameter values to configure BOOTP relay.

- The forward delay time.
- The hop count.
- The relay forwarding option.

The only parameter that is required for DHCP relay is the IP address to the DHCP server or to the next hop to the DHCP server. The default values can be accepted for forward delay, hop count, and relay forwarding option.

Alternately the relay function can be provided by an external router connected to the switch; in this case, the relay must be configured on the external router.

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 helper addresses. The following command sets the forward delay value of 10 seconds:

```
-> ip helper 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. The following syntax is used to set a maximum of four hops:

```
-> ip helper maximum-hops 4
```

The hops value represents the maximum number of relays. The default maximum hops value is set to four. This maximum hops value applies only to DHCP Relay. All other switch services ignore this value.

Setting the Relay Forwarding Option

This value specifies if DHCP Relay must operate in a Standard or Per VLAN only forwarding mode. By default, the forwarding option is set to standard. To change the forwarding option value, enter **ip helper** followed by **standard** or **per-vlan-only**. For example:

```
-> ip helper standard  
-> ip helper per-vlan-only
```

Using Automatic IP Configuration

An additional function of the DHCP Relay feature enables a switch to broadcast a BootP or DHCP request packet at boot time to obtain an IP address for default VLAN 1. This function is separate from the previously described functions (such as Global DHCP, per-VLAN DHCP, and related configurable options) in that enabling or disabling automatic IP configuration does not exclude or prevent other DHCP Relay functionality.

Note. Automatic IP address configuration only supports the assignment of a *permanent* IP address to the switch. Make sure that the DHCP server is configured with such an address before using this feature.

Using automatic IP configuration also allows the switch to specify the type of request packet to send; BootP or DHCP. When the BootP/DHCP server receives the request packet from the switch, it processes the request and sends an appropriate reply packet. When the switch receives a reply packet from the BootP/DHCP server, one or more of the following occurs:

- The router port for VLAN 1 is assigned the IP address provided by the server.
- If the reply packet contains a subnet mask for the IP address, the mask is applied to the VLAN 1 router port address. Otherwise, a default mask is determined based upon the class of the IP address. For example, if the IP address is a Class A, B, or C address, then 255.0.0.0, 255.255.0.0, or 255.255.255.0 is used for the subnet mask.
- If the reply packet from the server contains a gateway IP address, then a static route entry of 0.0.0.0 is created on the switch with the gateway address provided by the server.

Note. If the VLAN 1 router port is already configured with an IP address, the switch does not broadcast a request packet at boot time even if automatic IP configuration is enabled.

To verify IP router port configuration for VLAN 1, use the [show ip interface](#) and [show ip routes](#) commands. For more information about these commands, refer to the *OmniSwitch CLI Reference Guide*.

Enabling Automatic IP Configuration

By default, this function is disabled on the switch. To enable automatic IP configuration and specify the type of request packet, use the [ip helper boot-up](#) command. For example:

```
-> ip helper boot-up enable DHCP
-> ip helper boot-up enable BOOTP
```

The next time the switch boots up, DHCP Relay broadcasts a BootP request packet as the default or DHCP request packet if DHCP is enabled to obtain an IP address for default VLAN 1.

To disable automatic IP configuration for the switch, use the [ip helper boot-up](#) command with the **disable** option, as shown below:

```
-> ip helper boot-up disable
```

Configuring UDP Port 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 (NBNS/NBDD, other well-known UDP service ports, and service ports that are not well-known). This is done using UDP Port Relay commands to enable relay on these types of ports.

The UDP Port Relay function is separate from the previously described functions such as global DHCP, per-VLAN DHCP, and automatic IP configuration. Using UDP Port 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:

- You can configure either the **ip helper** or **ip udp relay service** but not both at the same time.
- The **ip udp relay service** command can also be used to enable or disable relay for DHCP well known ports 67 and 68.
- If the BOOTP/DHCP relay service is disabled, the **ip helper** configuration is *not* retained and all dependant functionalities, like automatic IP configuration for VLAN 1 and telnet, is disrupted.
- The DHCP Relay Agent Service must support the BOOTP/DHCP protocol, specifically UDP port 67 and 68.
- Use port 67 and/or port 68 with the **ip udp relay service** command to assign the well-known service BOOTP.
- Relaying DHCP traffic is available on a global and per-VLAN basis. Using this function on a per-VLAN basis requires setting the DHCP relay forwarding mode to **per-vlan-only**. UDP port relay for generic services is only available on a per-VLAN basis, however it does not require enabling the **per-vlan-only** forwarding option.

Configuring UDP Port Relay for generic UDP services is a two-step process. The first step involves enabling UDP Port Relay on the generic service port. The second step involves specifying a VLAN that relays and forwards the traffic destined for the generic service port. Both steps are required and are described below.

Enabling/Disabling UDP Port Relay

By default, a global relay operation is enabled for BOOTP/DHCP relay well-known ports 67 and 68 that becomes active when an IP network host address for a DHCP server is specified. To enable or disable a relay operation for a UDP service port, use the **ip udp relay service** command. For example, the following command enables relay on the DNS well-known service port:

```
-> ip udp relay service DNS
```

To enable relay on a user-defined UDP service port, enter the service port number using the **ip udp relay port** command. For example, the following command enables relay on service port 3047:

```
-> ip udp relay port 3047
```

To disable a relay operation for a UDP service port, use the **no** form of the **ip udp relay service** command. For example, the following command disables relay on the DNS well-known service port:

```
-> ip udp relay no service DNS
```

To disable a relay operation for a UDP service port, use the **no** form of the **ip udp relay port** command. For example, the following command disables relay on the DNS well-known service port:

```
-> ip udp relay no port 3047
```

For more information about using the **ip udp relay service** and **ip udp relay port** command, see the *OmniSwitch CLI Reference Guide*.

Specifying a Forwarding VLAN

To specify which VLAN(s) UDP Port Relay forwards traffic destined for a generic UDP service port, use the **ip udp relay service vlan** command. For example, the following command assigns VLAN 5 as a forwarding VLAN for the DNS well-known service port:

```
-> ip udp relay service dns vlan 5
```

Note. The **ip udp relay service vlan** command works only if UDP Port Relay is already enabled on the specified service port. In addition, when assigning a VLAN to the BOOTP/DHCP service ports, set the DHCP relay forwarding mode to **per-vlan-only** first before trying to assign the VLAN.

To specify more than one VLAN with a single command, enter a range of VLANs. For example, the following command assigns VLANs 6 through 8 as forwarding VLANs for the NBNS/NBDD well-known service ports:

```
-> ip udp relay service nbns vlan 6-8
```

If UDP Port Relay was enabled on a not well-known service port, then enter the service port number instead of the service name along with the **port** keyword. For example, the following command assigns VLAN 100 as a forwarding VLAN for UDP service port 3047:

```
-> ip 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 service vlan** command. For example, the following command removes the VLAN 6 association with the NBNS well-known service port:

```
-> no ip udp relay service nbns vlan 6
```

For more information about using the `ip udp relay service vlan` command, see the *OmniSwitch CLI Reference Guide*.

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 <code>ip helper agent-information policy</code> 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.

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 helper 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 helper agent-information policy drop
-> ip helper agent-information policy keep
-> ip helper 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.

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 CLI Reference Guide*. An example of the output for the **show ip helper** command is also given in “[Quick Steps for Setting Up DHCP Relay](#)” on page 17-4.

show ip helper	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 helper 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.

Note.

Use the **no ip helper statistics** command to reset the IP helper statistics for VRF instances.

Use the **ip udp relay no statistics** command to reset the generic UDP Relay Service related statistics.

18 Configuring VRRP

The Virtual Router Redundancy Protocol (VRRPv2/VRRPv3) is a standard router redundancy protocol supported in IPv4/IPv6, based on RFC 3768 and RFC 2787. It provides redundancy by eliminating the single point of failure inherent in a default route environment. The VRRPv2/VRRPv3 router, which controls the IPv4/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 Alcatel-Lucent implementation of VRRP also supports the collective management of virtual routers on a switch.

Note. This VRRPv3 implementation is based on the latest Internet Draft, Virtual Router Redundancy Protocol for IPv6, September 2004.

Note. 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 VRRPv2/VRRPv3 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 CLI Reference Guide*.

This chapter provides an overview of VRRP and includes information about the following:

- Virtual routers—see [“Creating/Deleting a Virtual Router”](#) on page 18-10.
- IP addresses for virtual routers—see [“Specifying an IP Address for a Virtual Router”](#) on page 18-11.
- VRRP advertisement interval—see [“Configuring the Advertisement Interval”](#) on page 18-12.
- Virtual router priority—see [“Configuring Virtual Router Priority”](#) on page 18-12.
- Preempting virtual routers—see [“Setting Preemption for Virtual Routers”](#) on page 18-12.
- VRRP traps—see [“Setting VRRP Traps”](#) on page 18-14.
- Configuring Collective Management Functionality—[“Configuring Collective Management Functionality”](#) on page 18-14
- Verifying the VRRP configuration—see [“Verifying the VRRP Configuration”](#) on page 18-18.
- VRRPv3 Virtual routers—see [“VRRPv3 Configuration Overview”](#) on page 18-19.
- IPv6 addresses for VRRPv3 virtual routers—see [“Specifying an IPv6 Address for a VRRPv3 Virtual Router”](#) on page 18-20.

- Accept mode for master router—see [“Configuring the VRRPv3 Advertisement Interval”](#) on page 18-21.
- VRRPv3 advertisement interval—see [“Configuring the VRRPv3 Advertisement Interval”](#) on page 18-21.
- VRRPv3 Virtual router priority—see [“Configuring the VRRPv3 Virtual Router Priority”](#) on page 18-21.
- Preempting VRRPv3 virtual routers—see [“Setting Preemption for VRRPv3 Virtual Routers”](#) on page 18-22.
- VRRPv3 traps—see [“Setting VRRPv3 Traps”](#) on page 18-23.
- VRRP tracking—see [“Creating Tracking Policies”](#) on page 18-25.
- VRRPv3 tracking—see [“Creating Tracking Policies”](#) on page 18-25.
- Verifying the VRRP configuration—see [“Verifying the VRRPv3 Configuration”](#) on page 18-24.

VRRP Specifications

Platforms Supported	OmniSwitch 10K
RFCs Supported	RFC 3768–Virtual Router Redundancy Protocol RFC 2787–Definitions of Managed Objects for the Virtual Router Redundancy Protocol
Compatible with HSRP	No
Maximum number of VRRPv2 and VRRPv3 virtual routers	255
Maximum number of IP addresses per instance	16

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 on off	Virtual routers are disabled (off)
Priority	priority	100
Preempt mode	preempt no preempt	Preempt mode is enabled
Advertising interval	advertising interval	1 second (100 centiseconds)

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	1 second (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
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	1
Default priority value for all the virtual routers in the group.	vrrp group	100

Default preempt mode for all the virtual routers in the group.	vrrp group	preempt
--	-------------------	----------------

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 traps	vrrp trap	Enabled
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 a VLAN ID. For example:

```
-> vrrp 6 4
```

The VLAN must already be created on the switch. For information about creating VLANs, see [Chapter 4, “Configuring VLANs.”](#)

- 2 Configure an IP address for the virtual router.

```
-> vrrp 6 4 address 10.10.2.3
```

Note. **IP** can be used as an optional parameter instead of **Address** in the above example.

- 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.

```
-> vrrp 6 4 enable
```

Note. *Optional.* To verify the VRRP configuration, enter the **show vrrp** command. The display is similar to the one shown here:

```
VRRP trap generation: Enabled
VRRP startup delay: 45 (expired)
      IP           Admin
VRID  VLAN  Address(es)  Status  Priority  Preempt  Interval
-----+-----+-----+-----+-----+-----+-----
  6    4    10.10.2.3   Enabled    100     yes      1
```

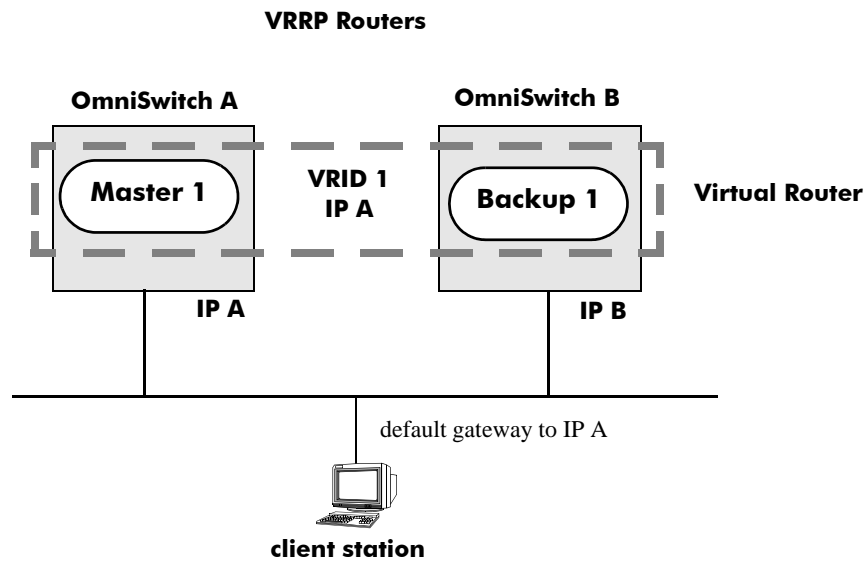
For more information about this display, see the *OmniSwitch CLI Reference Guide*.

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.



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 10K A is the master router because it contains the physical interface to which IP address A is assigned. OmniSwitch 10K 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 10K A will respond to ARP requests for IP address A using the virtual router's MAC address (00:00:5E:00:01:01) instead of the physical MAC address assigned to the interface. OmniSwitch 10K 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 ICMP ping requests).

OmniSwitch 10K 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 10K A becomes unavailable, OmniSwitch 10K B becomes the master router. OmniSwitch 10K B will then respond to ARP requests for IP address A using the virtual router's MAC address (00:00:5E:00:01:01). It will also forward packets for IP address B and respond to ARP requests for IP address B using the OmniSwitch 10K's physical MAC address.

OmniSwitch 10K B uses IP address B to access the LAN. However, IP address B is not backed up. Therefore, when OmniSwitch 10K 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 limitation of the OmniSwitch 10K is that a single VRID may be associated with a VLAN.

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:

$$\text{Master Down Interval} = (3 * \text{Advertisement Interval}) + \text{Skew Time}$$

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:

$$\text{Skew Time} = (256 - \text{Priority}) / 256$$

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 VRRPv2, and as the MAC address in neighbor advertisements sent by VRRPv3 (instead of the MAC address for the physical VRRP router).

The VRRPv2 (IPv4) address has the following format:

00-00-5E-00-01-[virtual router ID]

The VRRPv3 (IPv6) address has the following format:

00-00-5E-00-01-[virtual router ID]

This mapping provides for up to 255 virtual routers (VRRPv2 and VRRPv3 combined) on an OmniSwitch 10K.

ARP Requests

Each 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 10K 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.

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, 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 18-14](#).

Note. VRRP3 does not support the collective management functionality in this release.

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 VLAN 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 VLAN (either acting as master or backup), RDP will advertise all VLAN 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.

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. For more information about filtering, see [Chapter 21, “Configuring QoS.”](#)
- **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 statistics](#) 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 VLAN 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

To create a virtual router, enter the [vrrp](#) command with the desired VRID and the relevant VLAN ID. The VRID must be a unique number in the range from 1 to 255. The VLAN must already be created on the switch through the [vlan](#) command. For information about creating VLANs, see [Chapter 4, “Configuring VLANs.”](#) For example:

```
-> vrrp 6 4
```

This command creates VRID 6 on VLAN 4.

When you create a new virtual router, the VRID ID and a VLAN ID are *required*. Optionally, you may also specify:

- **Priority ([vrrp 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 18-12](#) for more information about how priority is used.

- **Preempt mode (vrrp preempt)**: 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 18-12.
- **Advertising interval (vrrp interval)**: Measured in seconds. Use the **interval** keyword with the desired number of seconds 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 18-12.

The following example creates a virtual router (with VRID 7) on VLAN 2 with a priority of 75. The preempt mode of the router is enabled and VRRP advertisements will be sent at intervals of 2 seconds:

```
-> vrrp 7 2 priority 75 preempt interval 2
```

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 enabled or disabled. *However, the virtual router must have an IP address assigned to it before it can be enabled.* 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 **vrrp** command with the relevant VRID and VLAN ID. For example:

```
-> no vrrp 7 3
```

Virtual router 7 on VLAN 3 is 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 CLI Reference Guide*.

Specifying an IP Address for a Virtual Router

An IP address must be specified before a virtual router may be enabled. To specify an IP address for a virtual router, use the **vrrp address** command and the relevant IP address. For example:

```
-> vrrp 6 4 address 10.10.2.3
-> vrrp 6 4 enable
```

In this example, the **vrrp address** command specifies that virtual router 6 on VLAN 4 will be used to backup IP address 10.10.2.3. The virtual router is then enabled with the **vrrp** command.

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.

To remove an IP address from a virtual router, use the **no** form of the **vrrp address** command. For example:

```
-> vrrp 6 4 disable
-> vrrp 6 4 no address 10.10.2.3
```

In this example, virtual router 6 is disabled. (A virtual router must be disabled before IP addresses may be added/removed from the router.) IP address 10.10.2.3 is then removed from the virtual router with the **no** form of the **vrrp address** command.

Configuring the Advertisement Interval

The advertisement interval is configurable, but all virtual routers with the same VRID must be configured with the same value. If the advertisement interval is set differently for a master router and a backup router, VRRP packets may be dropped because the newly configured interval does not match the interval indicated in the packet. The backup router will then take over and send a gratuitous ARP, which includes the virtual router IP address and the virtual router MAC address. In addition to creating duplicate IP/MAC address messages, both routers will begin forwarding packets sent to the virtual router MAC address. This will result in forwarding duplicate packets.

To avoid duplicate addresses and packets, make sure the advertisement interval is configured the same on both the master and the backup router.

For more information about VRRP and ARP requests, see “ARP Requests” on page 18-8.

To configure the advertisement interval, use the **vrrp** command with the **interval** keyword (**vrrp interval**). For example:

```
-> vrrp 6 4 disable
-> vrrp 6 4 interval 5
```

In this example, virtual router 6 is disabled. (If you are modifying an existing virtual router, the virtual router must be disabled before it may be modified.) The **vrrp** command is then used to set the advertising interval for virtual router 6 to 5 seconds. Optionally, you can also preface the **advertising** keyword before **interval**.

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. If a VRRP router owns the IP address of the virtual router and the IP address of the physical interface, this router will function as a virtual router master and its priority value will be 255. The value cannot be set to 255 if the router is not the IP address owner.

If there is more than one backup router, it is necessary to configure their priorities with different values. This is done so to elect the backup router with the highest value as the master. If the priority values are the same, the backup virtual router with the highest physical interface IP address is chosen as the master.

To set the priority, use the **vrrp** command with the **priority** keyword and the desired value (**vrrp priority**). For example:

```
-> vrrp 6 4 disable
-> vrrp 6 4 priority 50
```

In this example, virtual router 6 is disabled. (If you are modifying an existing virtual router, it must be disabled before it is modified.) The virtual router priority is then set to 50. Setting the value to 50 provides the router with a lower priority in the 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.

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 **vrrp** command (**vrrp preempt**). For example:

```
-> vrrp 6 4 disable
-> vrrp 6 4 no preempt
```

Note. The virtual router that owns the IP address(es) associated with the physical router always becomes the master router if it is available, regardless of the preempt mode setting and the priority values of the backup routers.

In the above example, the first command administratively disables virtual router 6. (If you are modifying an existing virtual router, it must be disabled before it is modified.). The second command disables the preempt mode for the same router. Henceforth, router 6 will not preempt another virtual router with a lower priority. For more information about priority, see [“Configuring Virtual Router Priority” on page 18-12](#).

Enabling/Disabling a Virtual Router

To enable a virtual router, use the **vrrp** command with the **enable** keyword. Note that at least one IP address must be configured for the virtual router through the **vrrp address** command. For example:

```
-> vrrp 7 3 priority 150
-> vrrp 7 3 address 10.10.2.3
-> vrrp 7 3 enable
```

In this example, a virtual router is created on VLAN 3 with a VRID of 7. An IP address is then assigned to the virtual router. The virtual router is then enabled on the switch.

To disable a virtual router, use the **disable** keyword.

```
-> vrrp 7 3 disable
```

A virtual router must be disabled before it may be modified. Use the **vrrp** command to disable the virtual router first; then use the command again to modify the parameters. For example:

```
-> vrrp 7 3 disable
-> vrrp 7 3 priority 200
-> vrrp 7 3 enable
```

In this example, virtual router 7 on VLAN 3 is disabled. The virtual router is then modified to change its priority setting. (For information about configuring the priority setting, see [“Configuring Virtual Router Priority” on page 18-12.](#)) The virtual router is 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 6 Switch Management Guide* for more information about enabling SNMP traps globally.

To disable VRRP traps, use the **no** form of the **vrrp trap** command.

```
-> no vrrp trap
```

To re-enable traps, enter the **vrrp trap** command.

```
-> vrrp trap
```

Setting VRRP Startup Delay

After a switch reboot, the delay which is a global value takes effect and all virtual routers 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 all the 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.

```
-> vrrp delay 75
```

The switch now waits 75 seconds after its reboot before it becomes available to take over as master for another router.

Note. This command applies only when the switch reboots.

Configuring Collective Management Functionality

Collective management simplifies the management and configuration tasks of either all the 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 virtual routers on a switch using the **vrrp interval** command. For example:

```
-> vrrp interval 50
```

You can change the default priority value of all the virtual routers on a switch using the **vrrp priority** command. For example:


```
-> vrrp priority 50
```

You can change the default preempt mode of all the virtual routers on a switch using the **vrrp preempt** command. For example:

```
-> vrrp no preempt
```

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 **vrrp set** command and enable the virtual routers again.

For example, to change the default priority value to 50 on all the existing virtual routers on a switch, enter the following:

```
-> vrrp priority 50
-> vrrp disable
-> vrrp set priority
-> vrrp 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 **vrrp set** command in this 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 **vrrp set** command along with the **override** option. For example:

```
-> vrrp set priority override
```

Note. You can specify a parameter such as interval, priority, preempt or all in 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 and preempt mode with the modified default values.

The next command 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 **vrrp** command along with the **enable all** option as follows:

```
-> vrrp 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.

A virtual router group can be created using the **vrrp group** command as follows:

```
-> vrrp group 25
```

This command creates a virtual router group 25. Use the **no** form of the same command to delete a virtual router group. For example:

```
-> no vrrp group 25
```

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 **vrrp group-association** command, as follows:

```
-> vrrp 10 1 group-association 25
```

The above command adds the virtual router 10 on VLAN 1 to the virtual router group 25. 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 reenabling the virtual router.

To remove a virtual router from a virtual router group, use the **no** form of the same command as follows:

```
-> vrrp 10 1 no group-association 25
```

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 like advertising interval, priority and preempt of all the virtual routers in a virtual router group using the **vrrp group** command, as follows:

```
-> vrrp group 25 advertising 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 **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 **vrrp group set** command and enable the virtual router group again. For example:

```
-> vrrp group 25 interval 50
-> vrrp group 25 disable
-> vrrp group 25 set interval
-> vrrp group 25 enable
```

The first command configures the default interval value as 50 for all the virtual routers in the virtual router group 25. The next command disables all the virtual routers in the group. **The vrrp group set** command in this sequence 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 **vrrp group set** command along with the **override** option. For example:

```
-> vrrp group set interval override
```

Note. You can specify a parameter such as interval, priority, preempt or all in 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 and preempt mode 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 **vrrp group** command with the **enable all** option as follows:

```
-> vrrp group 25 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.

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 CLI Reference Guide*.

VRRPv3 Configuration Overview

During startup, VRRPv3 is loaded onto the switch and is enabled. Virtual routers must be configured first and enabled as described in the sections. Since VRRPv3 is implemented on multiple switches in the network, some VRRPv3 parameters must be identical across switches:

- **VRRPv3 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. For more information about filtering, see [Chapter 21, “Configuring QoS.”](#)

- **Conflicting VRRPv3 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 statistics](#) command to check for conflicting parameters. For information about configuring VRRPv3 parameters, see the remaining sections of this chapter.

Basic VRRPv3 Virtual Router Configuration

At least two VRRPv3 virtual routers must be configured on the LAN—a master router and a backup router. The VRRPv3 virtual router is identified by a number called the Virtual Router ID (VRID), the VLAN on which the VRRPv3 virtual router is configured, and the IPv6 address or addresses associated with the router. Multiple VRRPv3 virtual routers may be configured on a single physical VRRP router.

Basic commands for setting up VRRPv3 virtual routers include:

```
vrrp3  
vrrp3 address
```

The next sections describe how to use these commands.

Creating/Deleting a VRRPv3 Virtual Router

To create a VRRPv3 virtual router, enter the [vrrp3](#) command with the desired VRID and the relevant VLAN ID. The VRID must be a unique number in the range from 1 to 255. The VLAN must already be created on the switch through the [vlan](#) command. For information about creating VLANs, see [Chapter 4, “Configuring VLANs.”](#) For example:

```
-> vrrp3 6 4
```

This command creates VRID 6 on VLAN 4.

When you create a new VRRPv3 virtual router, the VRID ID and a VLAN ID are *required*. Optionally, you may also specify:

- **Priority** ([vrrp 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 the VRRPv3 Virtual Router Priority” on page 18-21](#) for more information about how priority is used.

- **Preempt mode (vrrp preempt):** 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 VRRPv3 Virtual Routers](#)” on page 18-22.
- **Accept mode:** The **accept** mode allows the master router to accept packets addressed to the IPv6 address owner as its own. Use the **no accept** mode to prevent the master router from accepting packets addressed to the IPv6 address owner.
- **Advertising interval (vrrp interval):** Measured in seconds. Use the **interval** keyword with the desired number of centiseconds for the delay in sending VRRPv3 advertisement packets. You can change the default interval value and set a desired value. See “[Configuring the VRRPv3 Advertisement Interval](#)” on page 18-21.

Note. The maximum number of virtual routers supported is based on the 100 centisecond interval. A smaller interval will result in a relatively lesser number of virtual routers.

Note. The centisecond interval cannot be less than 10 centiseconds.

The following example creates a VRRPv3 virtual router (with VRID 7) on VLAN 2 with a priority of 75, and no preempt. VRRPv3 advertisements will be sent at intervals of 200 centiseconds:

```
-> vrrp3 7 2 priority 75 no preempt interval 200
```

Note. All VRRPv3 virtual routers with the same VRID on the same LAN should be configured with the same advertisement interval; otherwise the network may produce duplicate IPv6 or MAC address messages.

The **vrrp3** command may also be used to specify whether the VRRPv3 virtual router is enabled or disabled. For more information about the **vrrp3** command syntax, see the *OmniSwitch CLI Reference Guide*.

To delete a VRRPv3 virtual router, use the **no** form of the **vrrp3** command with the relevant VRID and VLAN ID. For example:

```
-> no vrrp3 7 3
```

VRRPv3 virtual router 7 on VLAN 3 is deleted from the configuration. (The virtual router does not have to be disabled before you delete it.)

Specifying an IPv6 Address for a VRRPv3 Virtual Router

A VRRPv3 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. Additional IPv6 addresses can be configured for a virtual router and these addresses must be within the subnet of an address configured on the interface. To specify an IPv6 address for a VRRPv3 virtual router, use the **vrrp3 address** command and the relevant IPv6 address. For example:

```
-> vrrp3 6 4 address fe80::200:5eff:fe00:20a
-> vrrp3 6 4 enable
```

In the above example, the **vrrp3 address** command specifies that VRRPv3 virtual router 6 on VLAN 4 will be used to backup IPv6 address `fe80::200:5eff:fe00:20a`. The virtual router is then enabled with the **vrrp3** command.

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 virtual router's link local address. In other words, a virtual router can not be the IP address owner if its link local address does not match the interface link local address.

To remove an IPv6 address from a virtual router, use the **no** form of the **vrrp3 address** command. For example:

```
-> vrrp3 6 4 disable
-> vrrp3 6 4 no address fe80::200:5eff:fe00:20a
```

In this example, VRRPv3 virtual router 6 is disabled. (A VRRPv3 virtual router must be disabled before IPv6 addresses may be added/removed from the router.) IP address `fe80::200:5eff:fe00:20a` is then removed from the virtual router with the **no** form of the **vrrp3 address** command.

Configuring the VRRPv3 Advertisement Interval

The advertisement interval is configurable, but all virtual routers with the same VRID must be configured with the same value. If the advertisement interval is set differently for a master router and a backup router, VRRPv3 packets may be dropped because the newly configured interval does not match the interval indicated in the packet. The backup router will then take over and send a neighbor advertisement, which includes the virtual router IP address and the virtual router MAC address. In addition to creating duplicate IP/MAC address messages, both routers will begin forwarding packets sent to the virtual router MAC address. This will result in forwarding duplicate packets.

To avoid duplicate addresses and packets, make sure the advertisement interval is configured the same on both the master and the backup router.

To configure the advertisement interval, use the **vrrp3** command with the **interval** keyword. For example:

```
-> vrrp3 6 4 disable
-> vrrp3 6 4 interval 500
```

In this example, VRRPv3 virtual router 6 is disabled. (If you are modifying an existing virtual router, the virtual router must be disabled before it may be modified.) The **vrrp3** command is then used to set the advertising interval for virtual router 6 to 500 centiseconds.

Configuring the VRRPv3 Virtual Router Priority

VRRPv3 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 IPv6 address; that is, the router contains the real physical interface to which the IPv6 address is assigned. The switch can change the default value and set it to 255 if it detects that the router is the IPv6 address owner. The value cannot be set to 255 if the router is not the IPv6 address owner.

The IPv6 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, any backup will take over for master.

Note that the switch sets the priority value to zero in the last VRRPv3 advertisement packet before a master router is disabled (see [“Enabling/Disabling a VRRPv3 Virtual Router” on page 18-23](#)).

Also, 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 **vrrp3** command with the **priority** keyword and the desired value. For example:

```
-> vrrp3 6 4 disable
-> vrrp3 6 4 priority 50
```

In this example, VRRPv3 virtual router 6 is disabled. (If you are modifying an existing virtual router, the virtual router must be disabled before it may 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 IPv6 address. Changing the default priority value and setting it to 50 would typically provide a router with lower priority in the VRRPv3 network.

Setting Preemption for VRRPv3 Virtual Routers

When a VRRPv3 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 the backup routers have priority values that are very nearly equal, the skew time may not be sufficient to overcome delays caused by network traffic loads and a lower priority backup may 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.

By default VRRPv3 virtual routers are allowed to preempt each other; that is, if 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 VRRPv3 virtual router that owns the IPv6 address(es) associated with the physical router always becomes the master router if it is available, regardless of the preempt mode setting and the priority values of the backup routers.

To disable preemption for a VRRPv3 virtual router, use the **vrrp3** command with the **no preempt** keywords. For example:

```
-> vrrp3 6 4 disable
-> vrrp3 6 4 no preempt
```

In this example, virtual router 6 is disabled. (If you are modifying an existing virtual router, the virtual router must be disabled before it may be modified.) The virtual router is then configured to disable preemption. If this virtual router takes over for an unavailable router, a router with a higher priority will not be able to preempt it. For more information about priority, see [“Configuring the VRRPv3 Virtual Router Priority” on page 18-21](#).

Enabling/Disabling a VRRPv3 Virtual Router

To change the default status of the VRRPv3 virtual router and to enable it, use the **vrrp3** command with the **enable** keyword. For example:

```
-> vrrp3 7 3
-> vrrp3 7 3 enable
```

In this example, a VRRPv3 virtual router is created on VLAN 3 with a VRID of 7. An IPv6 address is then assigned to the virtual router. The virtual router is then enabled on the switch.

To disable a VRRPv3 virtual router, use the **disable** keyword.

```
-> vrrp 7 3 disable
```

A VRRPv3 virtual router must be disabled before it may be modified. Use the **vrrp3** command to disable the virtual router first; then use the command again to modify the parameters. For example:

```
-> vrrp3 7 3 disable
-> vrrp3 7 3 priority 200
-> vrrp3 7 3 enable
```

In this example, VRRPv3 virtual router 7 on VLAN 3 is disabled. The VRRPv3 virtual router is then modified to change its priority setting. (For information about configuring the priority setting, see [“Configuring the VRRPv3 Virtual Router Priority” on page 18-21.](#)) The virtual router is then re-enabled and will be active on the switch.

Setting VRRPv3 Traps

A VRRPv3 router has the capability to generate VRRPv3 SNMP traps for events defined in the VRRPv3 SNMP MIB.

In order for VRRPv3 traps to be generated correctly, traps in general must be enabled on the switch through the SNMP CLI. See the *OmniSwitch AOS Release 6 Switch Management Guide* for more information about enabling SNMP traps globally.

To disable VRRPv3 traps, use the **no** form of the **vrrp3 trap** command.

```
-> no vrrp3 trap
```

To re-enable traps, enter the **vrrp3 trap** command:

```
-> vrrp3 trap
```

Verifying the VRRPv3 Configuration

A summary of the **show** commands used for verifying the VRRPv3 configuration is given here:

- | | |
|-------------------------------------|--|
| show vrrp3 | Displays the VRRPv3 virtual router configuration for all virtual routers or for a particular virtual router. |
| show vrrp3 statistics | Displays statistics about VRRPv3 packets for all VRRPv3 virtual routers configured on the switch or for a particular virtual router. |
| show vrrp3 track-association | Displays the tracking policies associated with VRRPv3 virtual routers. |

For more information about the displays that result from these commands, see the *OmniSwitch CLI Reference Guide*.

Creating 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:

```
-> vrrp track 3 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:

```
-> vrrp track 3 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.

Note the following:

- 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 VRRPv2/VRRPv3 Virtual Router

To associate a tracking policy with a virtual router, use the **vrrp track-association** command with the tracking policy ID number. In this example, virtual router 6 on VLAN 4 is disabled first so that tracking policy 3 may be associated with it:

```
-> vrrp 6 4 disable  
-> vrrp 6 4 track-association 3
```

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  
-> vrrp track 2 ipv4-interface vlan-4  
-> vrrp 5 4 track-association 2
```

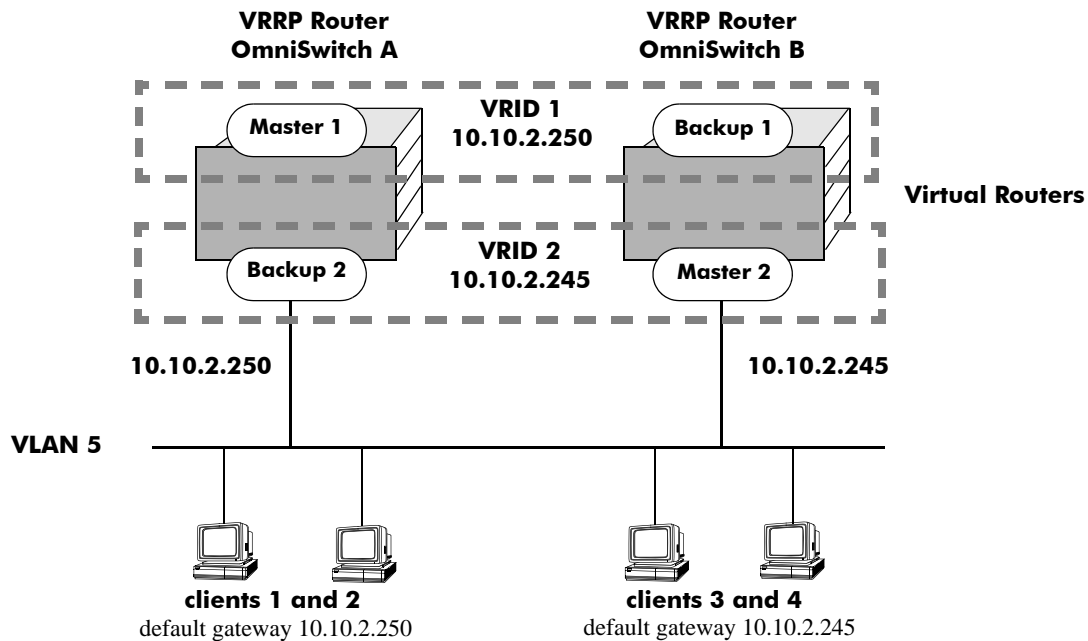
This configuration is allowed but will not really have an effect. If the associated interface goes down, this virtual router goes down as well and the tracking policy is not applied.

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.

VRRP Application Example

In addition to providing redundancy, 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).



VRRP Redundancy and Load Balancing

The CLI commands used to configure this setup are as follows:

1 First, create two virtual routers for VLAN 5. (Note that VLAN 5 must already be created and available on the switch.)

```
-> vrrp 1 5  
-> vrrp 2 5
```

2 Configure the IP addresses for each virtual router.

```
-> vrrp 1 5 ip 10.10.2.250  
-> vrrp 2 5 ip 10.10.2.245
```

3 Enable the virtual routers.

```
-> vrrp 1 5 enable  
-> vrrp 2 5 enable
```

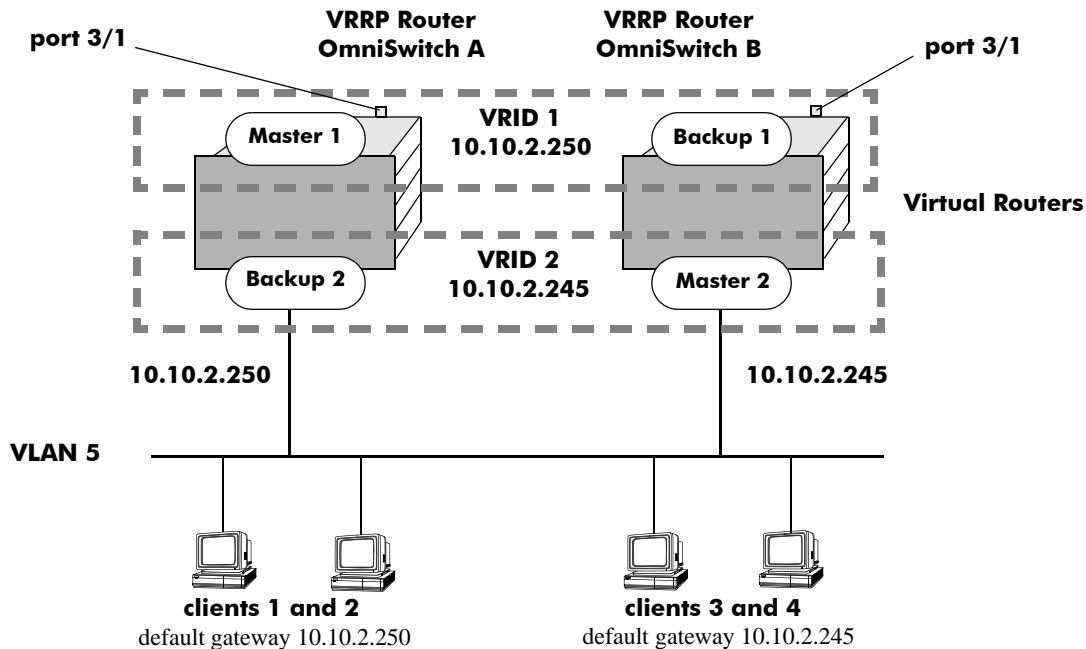
Note. The same VRRP configuration must be set up on each switch. The VRRP router that contains, or owns, the IP address will automatically become the master for that virtual router. If the IP address is a virtual address, the virtual router with the highest priority will become the master router.

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 10K 1 is the master for VRID 1 since it contains the physical interface to which 10.10.2.250 is assigned. If OmniSwitch 10K A should become unavailable, OmniSwitch 10K 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 10K B is the master for VRID 2 since it contains the physical interface to which 10.10.2.245 is assigned. If OmniSwitch 10K B should become unavailable, OmniSwitch 10K A will become master for 10.10.2.245. This configuration provides uninterrupted service for the end hosts.

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 10K A for clients 1 and 2 through IP address 10.10.2.250 and virtual router 2 serves as default gateway on OmniSwitch 10K 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 10K 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.



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:

```
-> vrrp 1 5 priority 100 preempt
-> vrrp 2 5 priority 75
```

The virtual router configuration for VRID 1 and 2 on VRRP router B is as follows:

```
-> vrrp 1 5 priority 75
-> vrrp 2 5 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.

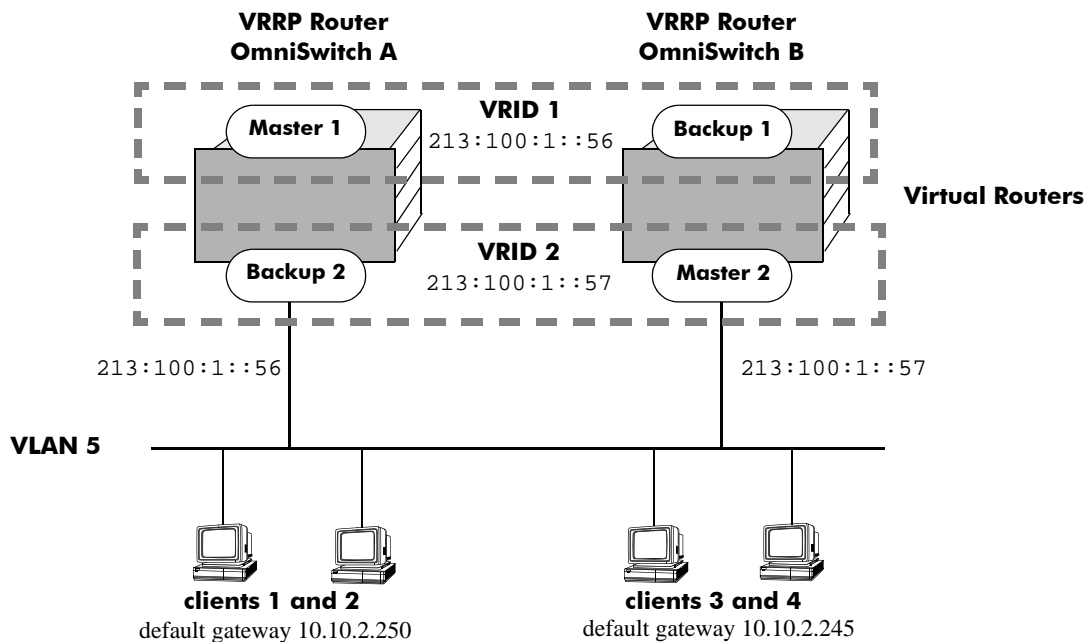
```
-> vrrp track 1 enable priority 50 port 3/1
-> vrrp 1 5 track-association 1
```

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 our 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 18-12](#) for more information about enabling preemption.

VRRPv3 Application Example

In addition to providing redundancy, VRRPv3 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).



VRRPv3 Redundancy and Load Balancing

The CLI commands used to configure this setup are as follows:

- 1 First, create two VRRPv3 virtual routers for VLAN 5. (Note that VLAN 5 must already be created and available on the switch.)

```
-> vrrp3 1 5
-> vrrp3 2 5
```

- 2 Configure the IPv6 addresses for each VRRPv3 virtual router.

```
-> vrrp3 1 5 address 213:100:1::56
-> vrrp3 2 5 address 213:100:1::57
```

- 3 Enable the VRRPv3 virtual routers.

```
-> vrrp3 1 5 enable
-> vrrp3 2 5 enable
```

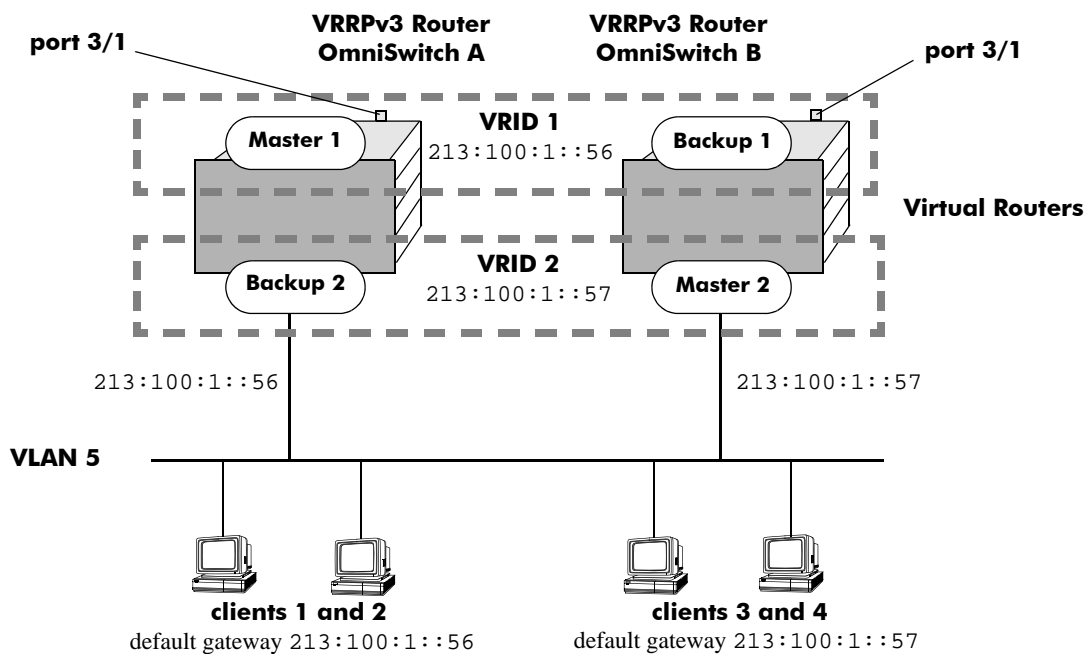

Note. The same VRRPv3 configuration must be set up on each switch. The VRRPv3 router that contains, or owns, the IPv6 address will automatically become the master for that virtual router. If the IPv6 address is a virtual address, the virtual router with the highest priority will become the master router.

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 10K 1 is the master for VRID 1 since it contains the physical interface to which 213:100:1::56s assigned. If OmniSwitch 10K A should become unavailable, OmniSwitch 10K 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 10K B is the master for VRID 2 since it contains the physical interface to which 213:100:1::57 is assigned. If OmniSwitch 10K B should become unavailable, OmniSwitch 10K A will become master for 213:100:1::57. This configuration provides uninterrupted service for the end hosts.

VRRPv3 Tracking Example

The figure below shows two VRRPv3 routers with two virtual routers backing up one IPv6 address on each VRRPv3 router respectively. Virtual router 1 serves as the default gateway on OmniSwitch 10K 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 10K 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.



VRRPv3 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 VRRPv3 router A is as follows:

```
-> vrrp3 1 5 priority 100 preempt
-> vrrp3 2 5 priority 75
```

The virtual router configuration for VRID 1 and 2 on VRRPv3 router B is as follows:

```
-> vrrp3 1 5 priority 75
-> vrrp3 2 5 priority 100 preempt
```

To ensure workstation clients 1 and 2 have connectivity to the internet, configure a tracking policy on VRRPv3 router A to monitor port 3/1 and associate the policy with VRID 1.

```
-> vrrp3 track 1 enable priority 50 port 3/1
-> vrrp3 1 5 track-association 1
```

If port 3/1 on VRR3 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 our 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 18-12](#) for more information about enabling preemption.

19 Configuring Server Load Balancing

Alcatel-Lucent's 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 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Procedures to configure SLB on a switch on [page 19-11](#).
- Procedures to configure logical SLB clusters on [page 19-12](#).
- Procedures to configure physical servers in SLB clusters on [page 19-14](#).
- Procedures to configure SLB probes on [page 19-18](#).
- Procedures for troubleshooting and maintenance on [page 19-17](#) and [page 19-22](#).

Note. You can also configure and monitor Server Load Balancing with WebView, Alcatel-Lucent's embedded web-based device management application. WebView is an interactive and easy-to-use GUI that can be launched from OmniVista or a web browser. Please refer to WebView's online documentation for more information on configuring and monitoring Server Load Balancing with WebView.

Server Load Balancing Specifications

The table below lists specifications for Alcatel-Lucent's SLB software.

Platforms Supported	OmniSwitch 10K
Maximum number of clusters	32
Maximum number of physical servers per cluster	32
Layer-3 classification	Destination IP address QoS policy condition
Layer-2 classification	QoS policy condition
Server health checking	Ping, link checks
High availability support	Hardware-based failover, VRRP, Chassis Management Module (CMM) redundancy
Networking protocols supported	Virtual IP (VIP) addresses
Maximum number of probes on a switch	40

Server Load Balancing Default Values

The table below lists default values for Alcatel-Lucent's SLB software.

Platforms Supported	OmniSwitch 10K
Maximum number of clusters	32
Maximum number of physical servers per cluster	32
Layer-3 classification	Destination IP address QoS policy condition
Layer-2 classification	QoS policy condition
Server health checking	Ping, link checks
High availability support	Hardware-based failover, VRRP, Chassis Management Module (CMM) redundancy
Networking protocols supported	Virtual IP (VIP) addresses
Maximum number of probes on a switch	40

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 19-5](#) 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 19-16 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

```

Cluster Name	Admin Status	Operational 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 c1
-> 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 19-7](#)), an SLB example ([“Server Load Balancing Example” on page 19-8](#)), and server health monitoring (see [“Server Health Monitoring” on page 19-10](#)).

Note. Alcatel-Lucent also offers link aggregation, which combines multiple Ethernet links into one virtual channel. Please refer to [Chapter 7, “Configuring Dynamic Link Aggregation,”](#) for more information on link aggregation and dynamic link aggregation, and to [Chapter 6, “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 19-12](#) 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:

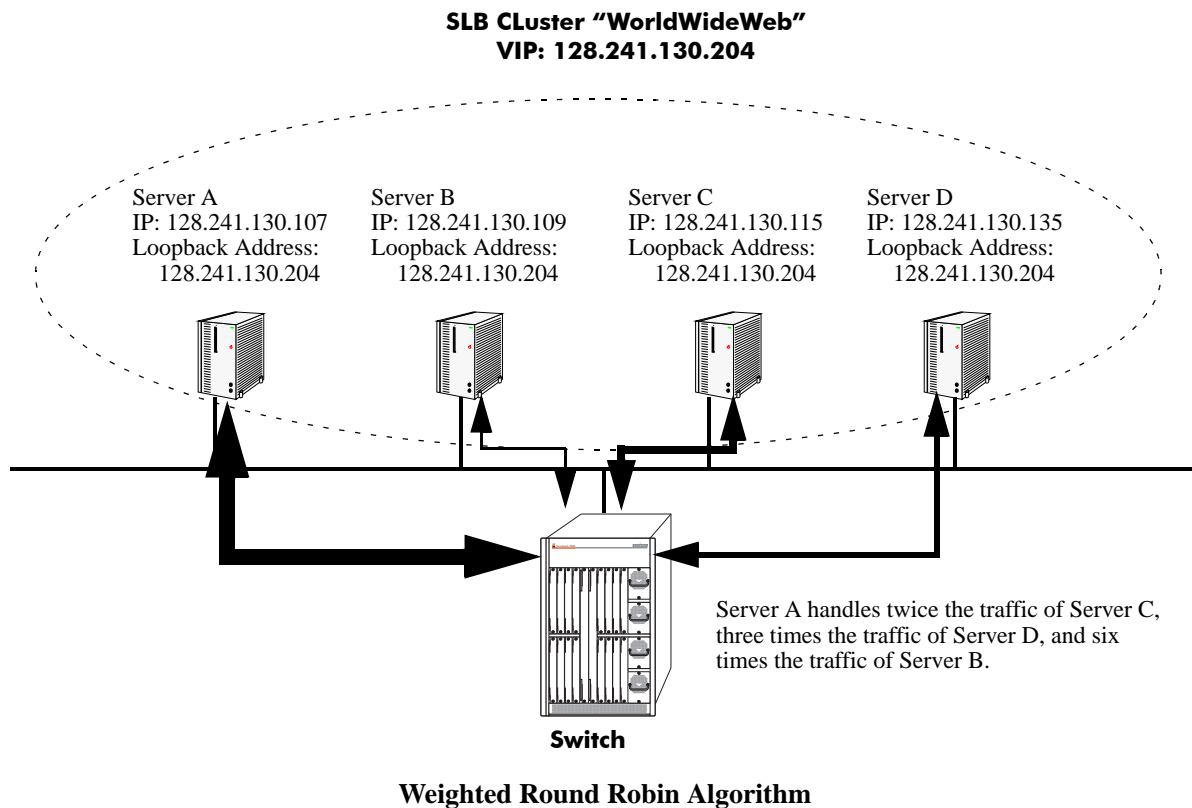
- The source IP address is changed to the IP address for the switch router interface.
- The destination IP address is changed to the IP address of the destined server.
- The TTL value is decremented.

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 19-12](#) for more information.

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 Alcatel-Lucent switch 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).



Note. See [“Modifying the Relative Weight of a Physical Server”](#) on page 19-16 for information on modifying the relative weights of servers in an SLB cluster.

Server Health Monitoring

Alcatel-Lucent's Server Load Balancing (SLB) software on the switch 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 19-22, 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 19-18 for more information.

Configuring Server Load Balancing on a Switch

This section describes how to use Alcatel-Lucent's 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 19-4](#) 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 19-11](#).
- 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 19-12](#).
- 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 19-14](#).

Note. Routing (which is enabled by default) must be enabled on a switch or Server Load Balancing will not operate.

Alcatel-Lucent's SLB software is preconfigured with the default values shown in the table in [“Server Load Balancing Default Values” on page 19-3](#). 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 19-15](#) 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

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
```

Note 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.

Configuring an SLB Cluster with a QoS Policy Condition

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
```

Note 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 be the switch configuration.

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 21, “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 21, “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 19-17).

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 19-3](#), Alcatel-Lucent's 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 19-15](#).
- **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 19-15](#).
- **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 19-16](#).
- **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 19-16](#).

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 Specifications" on [page 19-2](#) and "Server Load Balancing Default Values" on [page 19-3](#) tables.

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 19-15](#).

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 19-15](#).

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 19-14) and for administratively enabling and disabling a server in an SLB cluster (see [“Taking a Server On/Off Line”](#) on page 19-17).

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

Alcatel-Lucent's 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 19-22](#). 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 19-14](#)).

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 19-18 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 19-18 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 19-18 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 19-18 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
```

Cluster Name	VIP/COND	Admin Status	Operational 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 19-4](#) and [page 19-6](#) 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
mail_server	60000	3	3000	SMTP
mis_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 CLI Reference Guide* for complete syntax information on SLB **show** commands.

20 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 protocol (MLD). 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. Alcatel-Lucent's 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 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Enabling and disabling IP Multicast Switching and Routing on [page 20-8](#).
- Configuring and removing an IGMP static neighbor on [page 20-10](#).
- Configuring and removing an IGMP static querier on [page 20-11](#).
- Configuring and removing an IGMP static group on [page 20-11](#).
- Modifying IPMS parameters beginning on [page 20-13](#).
- Enabling and disabling IPv6 Multicast Switching and Routing on [page 20-23](#).
- Configuring and removing an MLD static neighbor on [page 20-25](#).
- Configuring and removing an MLD static querier on [page 20-26](#).
- Configuring and removing an MLD static group on [page 20-26](#).
- Modifying IPMSv6 parameters beginning on [page 20-28](#).

Note. You can also configure and monitor IPMS with WebView, Alcatel-Lucent's embedded Web-based device management application. WebView is an interactive and easy-to-use GUI that can be launched from OmniVista or a Web browser. Please refer to WebView's online documentation for more information on configuring and monitoring IPMS/IPMSv6 with WebView.

IPMS Specifications

The table below lists specifications for Alcatel-Lucent's IPMS software.

Platforms Supported	OS10K
RFCs Supported	RFC 1112 — Host Extensions for IP Multicasting RFC 2236 — Internet Group Management Protocol, Version 2 RFC 2933 — Internet Group Management Protocol MIB RFC 3376 — Internet Group Management Protocol, Version 3
IGMP Versions Supported	IGMPv1, IGMPv2, IGMPv3
Maximum number of IPv4/IPv6 forwarding entries	4K

IPMSv6 Specifications

The table below lists specifications for Alcatel-Lucent's IPMSv6 software.

RFCs Supported	RFC 2710 — Multicast Listener Discovery for IPv6 RFC 3019 — IPv6 MIB for Multicast Listener Discovery Protocol RFC 3810 — Multicast Listener Discovery Version 2 for IPv6
IETF Internet-Drafts Supported	draft-ietf-magma-snoop — Considerations for IGMP and MLD Snooping Switches
Platforms Supported	OS10K
MLD Versions Supported	MLDv1, MLDv2
MLD Query Interval	1 to 65535 in seconds
MLD Router Timeout	1 to 65535 in seconds
MLD Source Timeout	1 to 65535 in seconds
MLD Query Response Interval	1 to 65535 in milliseconds
MLD Last Member Query Interval	1 to 65535 in milliseconds

IPMS Default Values

The table below lists default values for Alcatel-Lucent's IPMS software.

Parameter Description	Command	Default Value/Comments
Administrative Status	ip multicast admin-state	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

IPMSv6 Default Values

The table below lists default values for Alcatel-Lucent's IPMSv6 software.

Parameter Description	Command	Default Value/Comments
Administrative Status	ip multicast helper-address	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

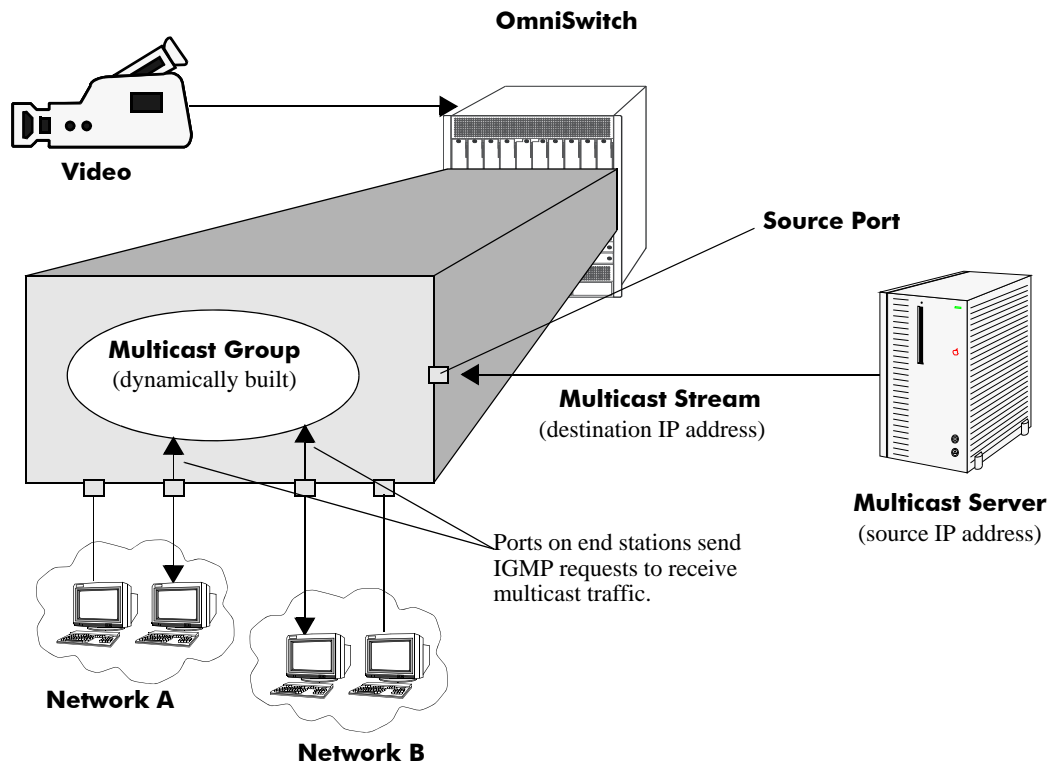
IPMS Overview

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 20-6 for more information.)

IPMS tracks the source VLAN 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.



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 20-7](#).
- Distance Vector Multicast Routing Protocol (DVMRP), which is described in [“DVMRP” on page 20-7](#).

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 7 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 20-10](#) 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.

Configuring IPMS on a Switch

This section describes how to use Command Line Interface (CLI) commands to enable and disable IP Multicast Switching and Routing (IPMSR) switch wide (see “[Enabling and Disabling IP Multicast Status](#)” on page 20-8), configure a port as a IGMP static neighbor (see “[Configuring and Removing an IGMP Static Neighbor](#)” on page 20-10), configure a port as a IGMP static querier (see “[Configuring and Removing an IGMP Static Querier](#)” on page 20-11), and configure a port as a IGMP static group (see “[Configuring and Removing an IGMP Static Group](#)” on page 20-11).

In addition, a tutorial is provided in “[IPMS Application Example](#)” on page 20-36 that shows how to use CLI commands to configure a sample network.

Note. See the “IP Multicast Switching Commands” chapter in the *OmniSwitch 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 configuration overrides the configuration of the system.

Enabling IP Multicast Status

To enable IP Multicast switching and routing on the system if no VLAN 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 by entering:

```
-> ip multicast vlan 2 admin-state enable
```

Disabling IP Multicast Status

To disable IP Multicast switching and routing on the system if no VLAN is specified, use the **ip multicast admin-state** command as shown below:

```
-> ip multicast admin-state disable
```

Or, as an alternative, enter:

```
-> ip multicast admin-state
```

To restore the IP Multicast status to its default setting.

You can also disable IP Multicast switching and routing on the specified VLAN by entering:

```
-> ip multicast vlan 2 admin-state disable
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 admin-state
```

To restore the IP Multicast status to its default setting.

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 is specified, you would enter:

```
-> ip multicast querier-forwarding enable
```

You can also enable the IGMP querier-forwarding on the specified VLAN by entering:

```
-> ip multicast vlan 2 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 is specified, you would enter:

```
-> ip multicast querier-forwarding disable
```

Or, as an alternative, enter:

```
-> ip multicast querier-forwarding
```

To restore the IGMP querier-forwarding to its default setting.

You can also disable the IGMP querier-forwarding on the specified VLAN by entering:

```
-> ip multicast vlan 2 querier-forwarding disable
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 querier-forwarding
```

To restore the IGMP querier-forwarding to its default setting.

You can remove an IGMP querier-forwarding entry on the specified VLAN and return to its default behavior by entering:

```
-> no ip multicast vlan 2 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 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 by entering:

```
-> ip multicast vlan 5 version 1
```

Restoring the IGMP Version

To restore the IGMP protocol version to its default version on the system if no VLAN is specified, use the **ip multicast version** command as shown below:

```
-> ip multicast version 0
```

Or, as an alternative, enter:

```
-> ip multicast version
```

To restore the IGMP version to its default version.

You can also restore the IGMP protocol version to version 2 on the specified VLAN by entering:

```
-> ip multicast vlan 2 version 0
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 version
```

To restore the IGMP version to its default version.

Configuring and Removing an IGMP Static Neighbor

IGMP static neighbor ports receive all multicast streams on the designated VLAN and also receive IGMP reports for the VLAN. The following subsections describe how to configure and remove a IGMP static neighbor port by using the **ip multicast max-group** command.

Configuring an IGMP Static Neighbor

You can configure a port as an IGMP static neighbor port by entering **ip multicast max-group** followed by **vlan**, a space, VLAN number (which must be between 0 and 4095), a space, followed by **port**, a space, the slot number of the port, a slash (/), and the port number.

For example, to configure port 10 in slot 4 with designated VLAN 2 as an IGMP static neighbor you would enter:

```
-> ip multicast static-neighbor vlan 2 port 4/10
```

You can also configure a link aggregation group as an IGMP static neighbor port by entering **ip multicast static-neighbor** followed by **vlan**, a space, VLAN number (which must be between 0 and 4095), a space, followed by **port**, a space, and the link aggregation group number.

For example, to configure link aggregation group 7 with designated VLAN 2 as a static neighbor you would enter:

```
-> ip multicast static-neighbor vlan 2 port 7
```

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 max-group** command by entering **no ip multicast static-neighbor** followed by **vlan**, a space, VLAN number, a space, followed by **port**, a space, the slot number of the port, a slash (/), and the port number.

For example, to remove port 10 in slot 4 with designated VLAN 2 as an IGMP static neighbor you would enter:

```
-> no ip multicast static-neighbor vlan 2 port 4/10
```

Configuring and Removing an IGMP Static Querier

IGMP static querier ports receive IGMP reports generated on the designated VLAN. 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

You can configure a port as an IGMP static querier port by entering **ip multicast static-querier**, followed by **vlan**, a space, the VLAN number (which must be between 0 and 4095), a space, followed by **port**, a space, the slot number of the port, a slash (/), and the port number.

For example, to configure port 10 in slot 4 with designated VLAN 2 as an IGMP static querier you would enter:

```
-> ip multicast static-querier vlan 2 port 4/10
```

You can also configure a link aggregation group as an IGMP static querier port by entering **ip multicast static-querier** followed by **vlan**, a space, VLAN number (which must be between 0 and 4095), a space, followed by **port**, a space, and the link aggregation group number.

For example, to configure link aggregation group 7 with designated VLAN 2 as a static querier you would enter:

```
-> ip multicast static-querier vlan 2 port 7
```

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 by entering **no ip multicast static-querier**, followed by **vlan**, a space, the VLAN number, a space, followed by **port**, a space, the slot number of the port, a slash (/), and the port number.

For example, to remove port 10 in slot 4 with designated VLAN 2 as an IPMS static querier you would enter:

```
-> no ip multicast static-querier vlan 2 port 4/10
```

Configuring and Removing an IGMP Static Group

IGMP static group ports receive IGMP reports generated on 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

You can configure a port as an IGMP static group by entering **ip multicast static-group**, followed by the IP address of the static group in dotted decimal notation, a space, followed by **vlan**, a space, VLAN number (which must be between 0 and 4095), a space, followed by **port**, a space, the slot number of the port, a slash (/), and the port number.

For example, to configure an IGMP static member with an IP address of 225.0.0.1 on port 10 in slot 3 with designated VLAN 3 you would enter:

```
-> ip multicast static-group 225.0.0.1 vlan 3 port 3/10
```

You can also configure a link aggregation group as an IPMS static group by entering **ip multicast static-group** followed by **vlan**, a space, VLAN number (which must be between 0 and 4095), a space, followed by **port**, a space, and the link aggregation group number.

For example, to configure link aggregation group 7 with designated VLAN 2 as a static group you would enter:

```
-> ip multicast static-group 225.0.0.2 vlan 2 port 7
```

Removing an IGMP Static Group

To reset the port so that it is no longer an IGMP static group port, use the **no** form of the **ip multicast static-group** command by entering **no ip multicast static-group**, followed by the IP address of the static group, a space, followed by **vlan**, a space, the VLAN number, a space, followed by **port**, the slot number of the port, a slash (/), and the port number.

For example, to remove an IGMP static member with an IP address of 225.0.0.1 on port 10 in slot 3 with designated VLAN 3 you would enter:

```
-> no ip multicast static-group 225.0.0.1 vlan 3 port 3/10
```

Modifying IPMS Parameters

The table in “[IPMS Default Values](#)” on page 20-3 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 in 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 in 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 is specified, you would enter:

```
-> ip multicast query-interval 60
```

You can also modify the IGMP query interval on the specified VLAN by entering:

```
-> ip multicast vlan 2 query-interval 60
```

Restoring the IGMP Query Interval

To restore the IGMP query interval to its default value on the system if no VLAN is specified, use the [ip multicast query-interval](#) command by entering:

```
-> ip multicast query-interval 0
```

Or, as an alternative, enter:

```
-> ip multicast query-interval
```

To restore the IGMP query interval to its default value.

You can also restore the IGMP query interval to its default value on the specified VLAN by entering:

```
-> ip multicast vlan 2 query-interval 0
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 query-interval
```

To restore the IGMP query interval to its default value.

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 in 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 in 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 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 by entering:

```
-> ip multicast vlan 3 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 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:

```
-> ip multicast last-member-query-interval
```

To restore the IGMP last member query interval to its default value.

You can also restore the IGMP last member query interval on the specified VLAN by entering:

```
-> ip multicast vlan 2 last-member-query-interval 0
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 last-member-query-interval
```

To restore the IGMP last member query interval to its default value.

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 in 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 you would enter:

```
-> ip multicast query-response-interval 6000
```

You can also modify the IGMP query response interval on the specified VLAN by entering:

```
-> ip multicast vlan 3 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 is specified, use the **ip multicast query-response-interval** command by entering:

```
-> ip multicast query-response-interval 0
```

Or, as an alternative, enter:

```
-> ip multicast query-response-interval
```

To restore the IGMP query response interval to its default value.

You can also restore the IGMP query response interval on the specified VLAN by entering:

```
-> ip multicast vlan 2 query-response-interval 0
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 query-response-interval
```

To restore the IGMP query response interval to its default value.

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 is specified, you would enter:

```
-> ip multicast router-timeout 360
```

You can also modify the IGMP router timeout on the specified VLAN by entering:

```
-> ip multicast vlan 2 router-timeout 360
```

Restoring the IGMP Router Timeout

To restore the IGMP router timeout to its default value on the system if no VLAN is specified, use the **ip multicast router-timeout** command by entering:

```
-> ip multicast router-timeout 0
```

Or, as an alternative, enter:

```
-> ip multicast router-timeout
```

To restore the IGMP router timeout to its default value.

You can also restore the IGMP router timeout on the specified VLAN by entering:

```
-> ip multicast vlan 2 router-timeout 0
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 router-timeout
```

To restore the IGMP router timeout to its default value.

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 router-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 is specified, you would enter:

```
-> ip multicast source-timeout 360
```

You can also modify the source timeout on the specified VLAN by entering:

```
-> ip multicast vlan 2 source-timeout 360
```

Restoring the Source Timeout

To restore the source timeout to its default value on the system if no VLAN is specified, use the [ip multicast source-timeout](#) command by entering:

```
-> ip multicast source-timeout 0
```

Or, as an alternative, enter:

```
-> ip multicast source-timeout
```

To restore the source timeout to its default value.

You can also restore the source timeout on the specified VLAN by entering:

```
-> ip multicast vlan 2 source-timeout 0
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 source-timeout
```

To restore the source timeout to its default value.

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
```

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:

```
-> ip multicast querying
```

To restore the IGMP querying to its default setting.

You can also disable the IGMP querying on the specified VLAN by entering:

```
-> ip multicast vlan 2 querying disable
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 querying
```

To restore the IGMP querying to its default setting.

You can remove an IGMP querying entry on the specified VLAN and return to its default behavior by entering:

```
-> 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 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 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 by entering:

```
-> ip multicast vlan 2 robustness 3
```

Restoring the IGMP Robustness Variable

You can restore the IGMP robustness variable to its default value on the system if no vlan is specified, by entering **ip multicast robustness** followed by the value 0 as shown below:

```
-> ip multicast robustness 0
```

Or, as an alternative, enter:

```
-> ip multicast robustness
```

To restore the IGMP robustness to its default value.

You can also restore the IGMP robustness variable to its default value on the specified VLAN, by entering **ip multicast robustness** followed by the value 0 as shown below:

```
-> ip multicast vlan 2 robustness 0
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 robustness
```

To restore the IGMP robustness to its default value.

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 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 by entering:

```
-> ip multicast vlan 2 spoofing enable
```

Disabling the IGMP Spoofing

To disable IGMP spoofing on the system if no VLAN is specified, use the **ip multicast spoofing** command as shown below:

```
-> ip multicast spoofing disable
```

Or, as an alternative, enter:

```
-> ip multicast spoofing
```

To restore the IGMP spoofing to its default setting.

You can also disable IGMP spoofing on the specified VLAN by entering:

```
-> ip multicast vlan 2 spoofing disable
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 spoofing
```

To restore the IGMP spoofing to its default setting.

You can remove an IGMP spoofing entry on the specified VLAN and return to its default behavior by entering:

```
-> no ip multicast vlan 2 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 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 by entering:

```
-> ip multicast vlan 2 zapping enable
```

Disabling the IGMP Zapping

To disable IGMP zapping on the system if no VLAN is specified, use the **ip multicast zapping** command as shown below:

```
-> ip multicast zapping disable
```

Or, as an alternative, enter:

```
-> ip multicast zapping
```

To restore the IGMP zapping to its default setting.

You can also disable IGMP zapping on the specified VLAN by entering:

```
-> ip multicast vlan 2 zapping disable
```

Or, as an alternative, enter:

```
-> ip multicast vlan 2 zapping
```

To restore the IGMP zapping to its default setting.

Limiting IGMP Multicast Groups

By default there is no limit on the number of IGMP groups that can be learned on a port/vlan instance. A maximum group limit can be set on a port, VLAN 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, or per port. Port settings override VLAN 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 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 and replace an existing session use the **ip multicast vlan max-group** command as shown below:

```
-> ip multicast vlan 10 max-group 25 action replace
```

To set the IGMP group limit for a port and drop any requests above the limit, use the **ip multicast port max-group** command as shown below:

```
-> ip multicast port 1/1 max-group 25 action drop
```

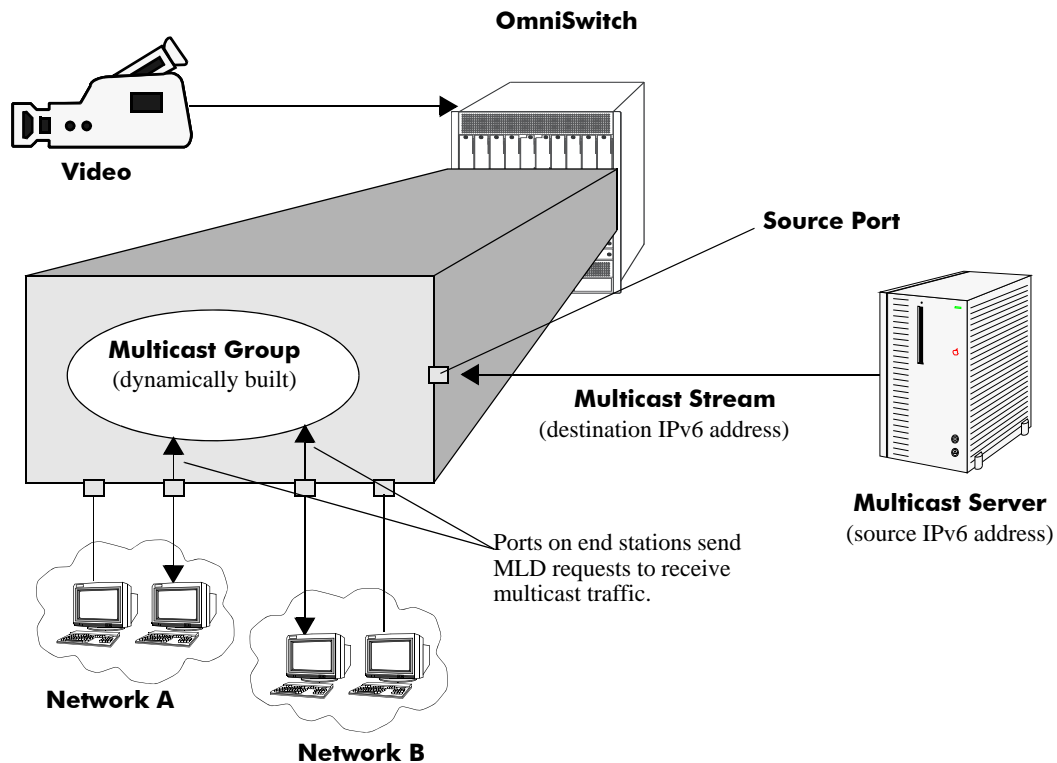
IPMSv6 Overview

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 20-22.)

IPMSv6 tracks the source VLAN 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.



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 20-24 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 enable and disable IPv6 Multicast Switching (IPMSv6) switch wide (see [“Enabling and Disabling IPv6 Multicast Status” on page 20-23](#)), configure a port as an MLD static neighbor (see [“Configuring and Removing an MLD Static Neighbor” on page 20-25](#)), configure a port as an MLD static querier (see [“Configuring and Removing an MLD Static Querier” on page 20-26](#)), and configure a port as an MLD static group (see [“Configuring and Removing an MLD Static Group” on page 20-26](#))

Note. See the “IP Multicast Switching Commands” chapter in the *OmniSwitch 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 [ip multicast helper-address](#) command.

Note. If IPv6 Multicast switching and routing is enabled on the system, the VLAN configuration overrides the configuration of the system.

Enabling IPv6 Multicast Status

To enable IPv6 Multicast switching and routing on the system if no VLAN is specified, use the [ip multicast helper-address](#) command as shown below:

```
-> ipv6 multicast admin-state enable
```

You can also enable IPv6 Multicast switching and routing on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 admin-state enable
```

Disabling IPv6 Multicast Status

To disable IPv6 Multicast switching and routing on the system if no VLAN is specified, use the [ip multicast helper-address](#) command as shown below:

```
-> ipv6 multicast admin-state disable
```

Or, as an alternative, enter:

```
-> ipv6 multicast admin-state
```

To restore the IPv6 Multicast status to its default setting.

You can also disable IPv6 Multicast on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 admin-state disable
```

Or, as an alternative, enter:

```
-> ipv6 multicast vlan 2 admin-state
```

To restore the IPv6 Multicast status to its default setting.

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 is specified, you would enter:

```
-> ipv6 multicast querier-forwarding enable
```

You can also enable the MLD querier-forwarding on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 querier-forwarding enable
```

Disabling the MLD Querier-forwarding

You can disable the MLD querier-forwarding by entering **ipv6 multicast querier-forwarding** followed by the **disable** keyword. For example, to disable the MLD querier-forwarding on the system if no VLAN is specified, you would enter:

```
-> ipv6 multicast querier-forwarding disable
```

Or, as an alternative, enter:

```
-> ipv6 multicast querier-forwarding
```

To restore the MLD querier-forwarding to its default setting.

You can also disable the MLD querier-forwarding on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 querier-forwarding disable
```

Or, as an alternative, enter:

```
-> ipv6 multicast vlan 2 querier-forwarding
```

To restore the MLD querier-forwarding to its default setting.

You can remove an MLD querier-forwarding entry on the specified VLAN and return to its default behavior by entering:

```
-> no ipv6 multicast vlan 2 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 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 is specified, use the **ipv6 multicast version** command by entering:

```
-> ipv6 multicast version 0
```

Or, as an alternative, enter:

```
-> ipv6 multicast version
```

To restore the MLD version to Version 1.

You can also restore the MLD version to Version 1 (MLDv1) on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 version 0
```

Or, as an alternative, enter:

```
-> ipv6 multicast vlan 2 version
```

To restore the MLD version to Version 1.

Configuring and Removing an MLD Static Neighbor

MLD static neighbor ports receive all multicast streams on the designated VLAN and also receive MLD reports for the VLAN. The following subsections describe how to configure and remove a static neighbor port by using the **ipv6 multicast max-group** command.

Configuring an MLD Static Neighbor

You can configure a port as an MLD static neighbor port by entering **ipv6 multicast max-group** followed by **vlan**, a space, VLAN number (which must be between 0 and 4095), a space, followed by **port**, a space, the slot number of the port, a slash (/), and the port number.

For example, to configure port 10 in slot 4 with designated VLAN 2 as an MLD static neighbor you would enter:

```
-> ipv6 multicast static-neighbor vlan 2 port 4/10
```

You can also configure a link aggregation group as an MLD static neighbor port by entering **ipv6 multicast static-neighbor** followed by **vlan**, a space, VLAN number (which must be between 0 and 4095), a space, followed by **port**, a space, and the link aggregation group number.

For example, to configure link aggregation group 7 with designated VLAN 2 as a static neighbor you would enter:

```
-> ipv6 multicast static-neighbor vlan 2 port 7
```

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 max-group** command by entering **no ipv6 multicast static-neighbor**, followed by **vlan**, a space, the VLAN number, a space, followed by **port**, a space, slot number of the port, a slash (/), and the port number.

For example, to remove port 10 in slot 4 with designated VLAN 2 as an MLD static neighbor you would enter:

```
-> no ipv6 multicast static-neighbor vlan 2 port 4/10
```

Configuring and Removing an MLD Static Querier

MLD static querier ports receive MLD reports generated on the designated VLAN. 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

You can configure a port as an MLD static querier port by entering **ipv6 multicast static-querier**, followed by **vlan**, a space, the VLAN number (which must be between 0 and 4095), a space, followed by **port**, a space, the slot number of the port, a slash (/), and the port number.

For example, to configure port 10 in slot 4 with designated VLAN 2 as an MLD static querier you would enter:

```
-> ipv6 multicast static-querier vlan 2 port 4/10
```

You can also configure a link aggregation group as an MLD static querier port by entering **ipv6 multicast static-querier**, followed by **vlan**, a space, VLAN number (which must be between 0 and 4095), a space, followed by **port**, a space, and the link aggregation group number.

For example, to configure link aggregation group 7 with designated VLAN 2 as a static querier you would enter:

```
-> ipv6 multicast static-querier vlan 2 port 7
```

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 by entering **no ipv6 multicast static-querier**, followed by **vlan**, a space, the VLAN number, a space, followed by **port**, a space, the slot number of the port, a slash (/), and the port number.

For example, to remove port 10 in slot 4 with designated VLAN 2 as a static querier you would enter:

```
-> no ipv6 multicast static-querier vlan 2 port 4/10
```

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

You can configure a port as an MLD static group by entering **ipv6 multicast static-group**, followed by the IPv6 address of the MLD static group in hexadecimal notation separated by colons, a space, followed by **vlan**, a space, VLAN number (which must be between 0 and 4095), a space, followed by **port**, the slot number of the port, a slash (/), and the port number.

For example, to configure an MLD static group with an IPv6 address of `ff05::5` enter:

```
-> ipv6 multicast static-group ff05::5 vlan 3 port 3/10
```

You can also configure a link aggregation group as an MLD static group by entering **ipv6 multicast static-group**, followed by **vlan**, a space, VLAN number (which must be between 0 and 4095), a space, followed by **port**, a space, and the link aggregation group number.

For example, to configure link aggregation group 7 with designated VLAN 2 as a static group you would enter:

```
-> ipv6 multicast static-group ff05::6 vlan 2 port 7
```

Removing an MLD Static Group

To reset the port so that it is no longer an MLD static group port, use the **no** form of the **ipv6 multicast static-group** command by entering **no ipv6 multicast static-group**, followed by the IPv6 address of the static group in hexadecimal notation separated by colons, a space, followed by **vlan**, a space, VLAN number, a space, followed by **port**, a space, the slot number of the port, a slash (/), and the port number.

For example, to remove an MLD static member with an IPv6 address of `ff05::5` on port 10 in slot 3 with designated VLAN 3 you would enter:

```
-> no ipv6 multicast static-group ff05::5 vlan 3 port 3/10
```

Modifying IPMSv6 Parameters

The table in “[IPMSv6 Default Values](#)” on page 20-4 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 in 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 is specified, you would enter:

```
-> ipv6 multicast query-interval 160
```

You can also modify the MLD query interval on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 query-interval 160
```

Restoring the MLD Query Interval

To restore the MLD query interval to its default value on the system if no VLAN is specified, use the [ipv6 multicast query-interval](#) command by entering:

```
-> no ipv6 multicast query-interval
```

You can also restore the MLD query interval on the specified VLAN by entering:

```
-> no ipv6 multicast vlan 2 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 in 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 in 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 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 by entering:

```
-> ipv6 multicast vlan 3 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 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:

```
-> ipv6 multicast last-member-query-interval
```

To restore the MLD last member query interval to its default value.

You can also restore the MLD last member query interval on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 last-member-query-interval 0
```

Or, as an alternative, enter:

```
-> ipv6 multicast vlan 2 last-member-query-interval
```

To restore the MLD last member query interval to its default value.

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 in 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 in 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 by entering:

```
-> ipv6 multicast vlan 3 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 is specified, use the **ipv6 multicast query-response-interval** command by entering:

```
-> ipv6 multicast query-response-interval 0
```

Or, as an alternative, enter:

```
-> ipv6 multicast query-response-interval
```

To restore the MLD query response interval to its default value.

You can also restore the MLD query response interval on the specified VLAN by entering:

```
-> ipv6 multicast van 2 query-response-interval 0
```

Or, as an alternative, enter:

```
-> ipv6 multicast vlan 2 query-response-interval
```

To restore the MLD query response interval to its default value.

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 is specified, you would enter:

```
-> ipv6 multicast router-timeout 360
```

You can also modify the MLD router timeout on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 router-timeout 360
```

Restoring the MLD Router Timeout

To restore the MLD router timeout to its default value on the system if no VLAN is specified, use the [ipv6 multicast router-timeout](#) command by entering:

```
-> ipv6 multicast router-timeout 0
```

Or, as an alternative, enter:

```
-> ipv6 multicast router-timeout
```

To restore the MLD router timeout to its default value.

You can also restore the MLD router timeout on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 router-timeout 0
```

Or, as an alternative, enter:

```
-> ipv6 multicast vlan 2 router-timeout
```

To restore the MLD router timeout to its default value.

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 is specified, you would enter:

```
-> ipv6 multicast source-timeout 60
```

You can also modify the source timeout on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 source-timeout 60
```

Restoring the Source Timeout

To restore the source timeout to its default value on the system if no VLAN is specified, use the **ipv6 multicast source-timeout** command by entering:

```
-> ipv6 multicast source-timeout 0
```

Or, as an alternative, enter:

```
-> ipv6 multicast source-timeout
```

To restore the source timeout to its default value.

You can also restore the source timeout on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 source-timeout 0
```

Or, as an alternative, enter:

```
-> ipv6 multicast vlan 2 source-timeout
```

To restore the source timeout to its default value.

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
```

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:

```
-> ipv6 multicast querying
```

To restore the MLD querying to its default setting.

You can also disable the MLD querying on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 querying disable
```

Or, as an alternative, enter:

```
-> ipv6 multicast vlan 2 querying
```

To restore the MLD querying to its default setting.

You can remove an MLD querying entry on the specified VLAN and return to its default behavior by entering:

```
-> 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 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 by entering:

```
-> ipv6 multicast vlan 2 robustness 3
```

Restoring the MLD Robustness Variable

You can restore the MLD robustness variable to its default value on the system if no vlan is specified by entering **ipv6 multicast robustness** followed by the value 0, as shown below:

```
-> ipv6 multicast robustness 0
```

Or, as an alternative, enter:

```
-> ipv6 multicast robustness
```

To restore the MLD robustness to its default value.

You can also modify the MLD robustness variable from 1 to 7 on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 robustness 0
```

Or, as an alternative, enter:

```
-> ipv6 multicast vlan 2 robustness
```

To restore the MLD robustness to its default value.

Enabling and Disabling the 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 the MLD Spoofing

To enable MLD spoofing on the system if no VLAN 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 by entering:

```
-> ipv6 multicast vlan 2 spoofing enable
```

Disabling the MLD Spoofing

To disable MLD spoofing on the system if no VLAN is specified, you use the [ipv6 multicast spoofing](#) command as shown below:

```
-> ipv6 multicast spoofing disable
```

Or, as an alternative, enter:

```
-> ipv6 multicast spoofing
```

To restore the MLD spoofing to its default setting.

You can also disable MLD spoofing on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 spoofing disable
```

Or, as an alternative, enter:

```
-> ipv6 multicast vlan 2 spoofing
```

To restore the MLD spoofing to its default setting.

You can remove an MLD spoofing entry on the specified VLAN and return to its default behavior by entering:

```
-> no ipv6 multicast vlan 2 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 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 by entering:

```
-> ipv6 multicast vlan 2 zapping enable
```

Disabling the MLD Zapping

To disable MLD zapping on the system if no VLAN is specified, use the **ipv6 multicast zapping** command as shown below:

```
-> ipv6 multicast zapping disable
```

Or, as an alternative, enter:

```
-> ipv6 multicast zapping
```

To restore the MLD zapping to its default setting.

You can also disable MLD zapping on the specified VLAN by entering:

```
-> ipv6 multicast vlan 2 zapping disable
```

Or, as an alternative, enter:

```
-> ipv6 multicast vlan 2 zapping
```

To restore the MLD zapping to its default setting.

Limiting MLD Multicast Groups

By default there is no limit on the number of MLD groups that can be learned on a port/vlan instance. A maximum group limit can be set on a port, VLAN 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, or per port. Port settings override VLAN 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 **ip multicast max-group** command as shown below:

```
-> ipv6 multicast max-group 25 action drop
```

To set the MLD group limit for a VLAN and replace any requests above the limit, use the **ip multicast vlan max-group** command as shown below:

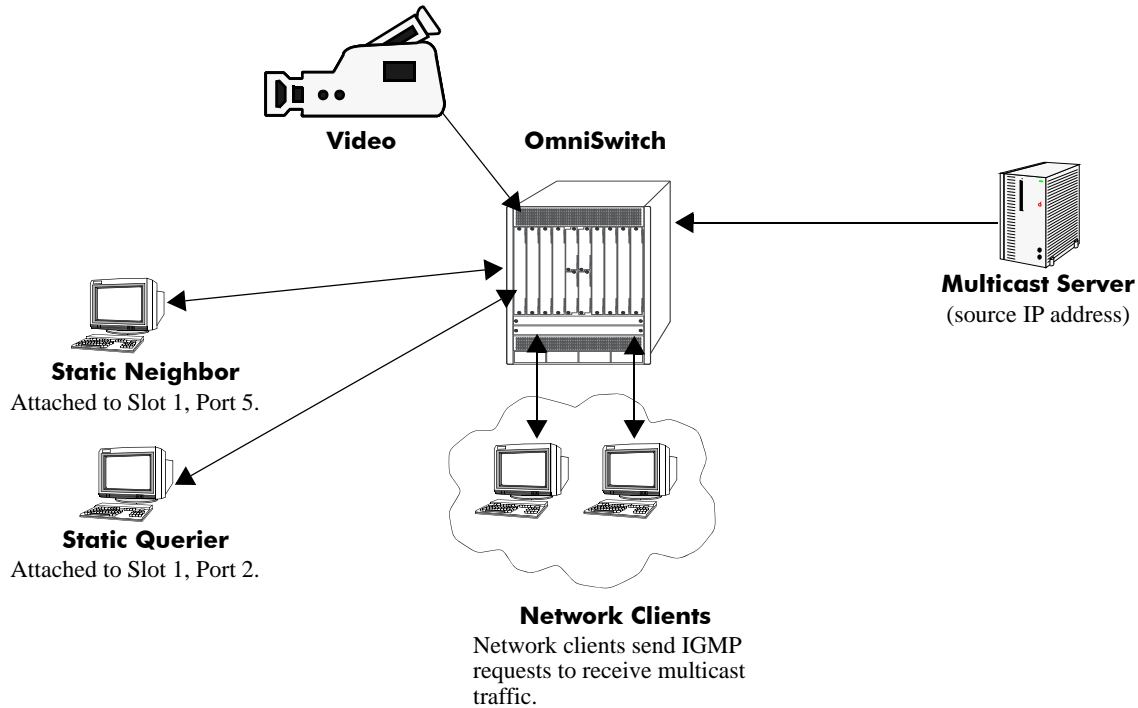
```
-> ipv6 multicast vlan 10 max-group 25 action replace
```

To set the MLD group limit for a port and drop any requests above the limit, use the **ip multicast port max-group** command as shown below:

```
-> ipv6 multicast port 1/1 max-group 25 action drop
```

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.



Example of 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
```

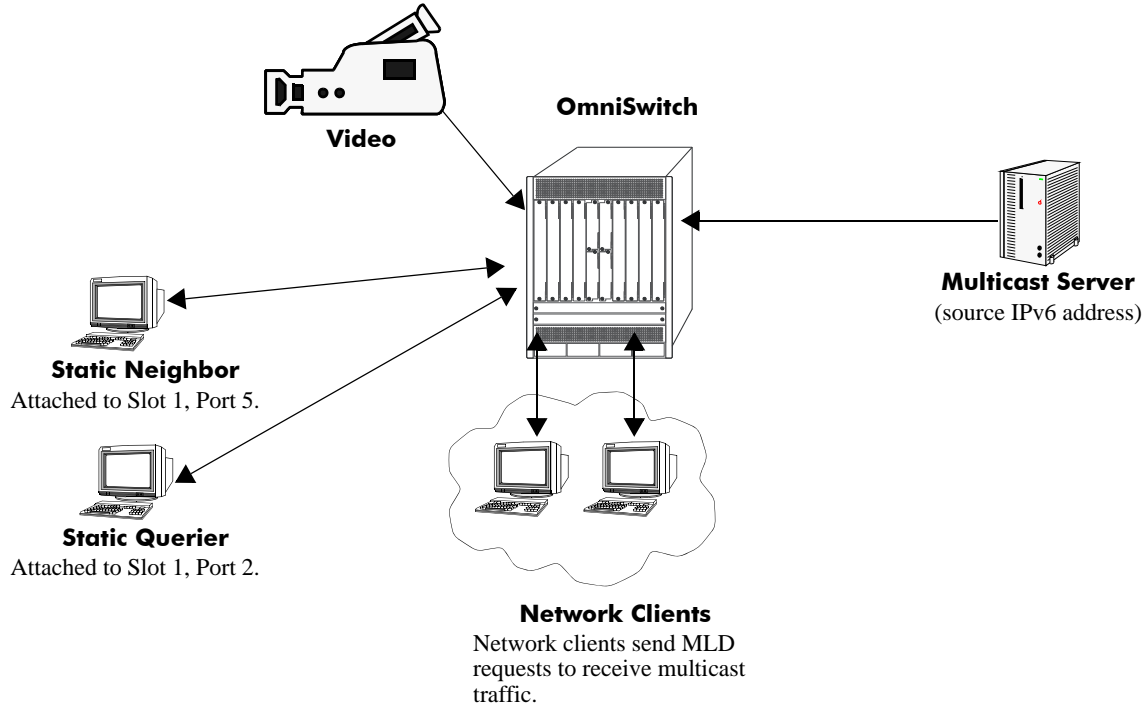
```
Total 1 Neighbors
Host Address    VLAN  Port  Static  Count  Life
-----+-----+-----+-----+-----
1.0.0.2        5    1/5   no      1      86
```

```
-> 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.



Example of IMPS 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

Alcatel-Lucent's 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	Displays the IPMS multicast source table. If you do not specify a multicast group IP address, then the source table 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.

If you are interested in a quick look at IPMS groups on your switch you could use the **show ip multicast group** command. For example:

```
-> show ip multicast group
```

```
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
```

Note. See the “IP Multicast Switching Commands” chapter in the *OmniSwitch CLI Reference Guide* for complete documentation on IPMS **show** commands.

Displaying IPMSv6 Configurations and Statistics

Alcatel-Lucent's 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	Displays the IPMSv6 multicast source table. If you do not specify a multicast group IPv6 address, then the source 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.

If you are interested in a quick look at IPMSv6 groups on your switch you could use the **show ipv6 multicast group** command. For example:

```
-> show ipv6 multicast group
```

```
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
```

Note. See the “IPv6 Multicast Switching Commands” chapter in the *OmniSwitch CLI Reference Guide* for complete documentation on IPMS **show** commands.

21 Configuring QoS

The OmniSwitch 10K QoS software and embedded virtual output queue (VOQ) 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 prioritization 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.
- **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 21-59](#).

The implementation of VOQ integrates traffic management with QoS scheduling. Embedded profiles apply the QoS admission control and bandwidth management configurations to VOQ flows. Packets received by the switch are queued on the ingress to avoid congestion on the egress. A centralized scheduler in the switch fabric arbitrates flows between ingress and egress ports based on feedback from the egress port.

In This Chapter

This chapter describes QoS in general and how policies, port-based QoS configuration, and VOQ 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 CLI Reference Guide*.

The following topics and procedures are included in this chapter:

- [“QoS General Overview” on page 21-4](#).
- [“Classification” on page 21-6](#).
- [“Congestion Management” on page 21-10](#).
- [“Congestion Avoidance” on page 21-18](#).
- [“Traffic Policing and Shaping” on page 21-20](#).

-
- “QoS Defaults” on page 21-30.
 - “Configuring QoS” on page 21-34.
 - “Creating Policies” on page 21-39.
 - “Using Access Control Lists” on page 21-59.
 - “Policy Applications” on page 21-69.

QoS Specifications

The QoS functionality described in this chapter is supported on the OmniSwitch 10K switches, unless otherwise stated in the following QoS Specifications table or specifically noted within any other section of this chapter. Note that any maximum limits provided in the Specifications table are subject to available system resources.

Maximum number of policy rules	8192
Maximum number of policy conditions	8192
Maximum number of policy actions	8192
Maximum number of validity periods	128
Maximum number of policy services	512
Maximum number of groups (network, MAC, service, port)	2048
Maximum number of group entries	1024 per group (512 per service group)
Maximum number of rules per slot	6144 (1152 on OS10K-XNI-U32S modules)
Maximum number of Class of Service (CoS) virtual output queues (VOQ) per port.	8
CLI Command Prefix Recognition	Some QoS commands support prefix recognition. See the “Using the CLI” chapter in the <i>OmniSwitch AOS Release 7 Switch Management Guide</i> for more information.

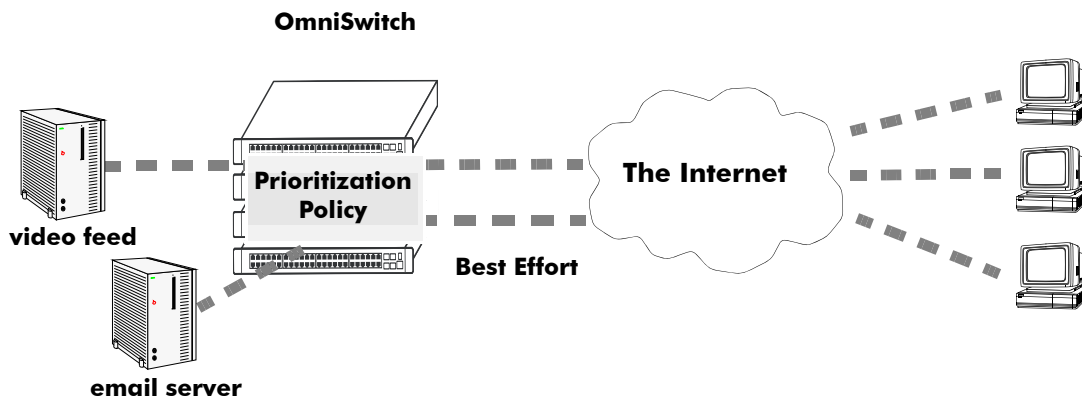
QoS General Overview

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.



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 OmniSwitch 10K integrates traffic management with QoS scheduling. A virtual output queue (VOQ) architecture is implemented to funnel traffic from ingress to egress, requiring only one stage of packet buffering. Deep ingress buffers facilitate holding ingress traffic when congestion occurs on the egress, helping to provide delivery with little or no packet loss.

The general order of events with respect to the OmniSwitch 10K 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 21-6](#) for more information.

2 Congestion Management—Classified packets are prioritized and funneled from ingress to egress using queuing technologies, such as Class of Service (CoS) VOQs, to ensure preferential treatment to high priority traffic. See [“Congestion Management” on page 21-10](#).

3 Congestion Avoidance—Weighted Random Early Detection (WRED) is used for admission control and bandwidth management. Packets that are not high priority are randomly dropped to help avoid “tail drop” on the queues. See [“Congestion Avoidance” on page 21-18](#).

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 21-20](#).

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, queuing 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 10K virtual output queuing (VOQ) capability uses these CoS values to determine the forwarding treatment by prioritizing flows based on application and network requirements. For more information about VOQ and queue (congestion) management, see [“Congestion Management” on page 21-10](#).

How Traffic is Classified and Marked

The OmniSwitch 10K 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 21-26](#) 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 21-59](#) for more information.

- **Port-based QoS**

Individual ports are configured to either trust (recognize) or not trust (don't 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 21-8](#) 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 21-60](#) 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 21-7](#) 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 21-59](#) 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 detected by examining the source MAC address of the packet to determine if the address falls within the following ranges of IP phone MAC addresses:

00-80-9F-54-xx-xx to 00-80-9F-64-xx-xx
00-80-9F-66-xx-xx to 00-80-9F-6F-xx-xx.

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 “alaPhone” MAC address group. See [“Creating MAC Groups” on page 21-52](#) 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 CLI Reference Guide* for more information.

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 trust setting for a port can be configured globally on the switch 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 trusted
```

The global setting is active immediately; however, the port setting requires **qos apply** to activate the change. For more information about the **qos apply** command, see [“Applying the Configuration” on page 21-66](#).

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/ Port	Active	Trust	Default P/DSCP	Default Classification	Physical	Bandwidth Ingress	Egress	Type
1/1	No	No	0/ 0	DSCP	0K	-	-	ethernet
1/2	Yes	No	0/ 0	DSCP	1.00G	-	-	ethernet-1G
1/3	No	No	0/ 0	DSCP	0K	-	-	ethernet
1/4	No	No	0/ 0	DSCP	0K	-	-	ethernet
1/5	No	No	0/ 0	DSCP	0K	-	-	ethernet
1/6	No	No	0/ 0	DSCP	0K	-	-	ethernet
1/7	No	No	0/ 0	DSCP	0K	-	-	ethernet
1/8	No	No	0/ 0	DSCP	0K	-	-	ethernet
1/9	No	No	0/ 0	DSCP	0K	-	-	ethernet
1/10	No	No	0/ 0	DSCP	0K	50K	-	ethernet
1/11	No	*Yes	0/ 0	*802.1P	0K	-	-	ethernet
1/12	No	*Yes	0/ 0	*802.1P	0K	-	-	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 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. For more information about the **qos apply** command, see [“Applying the Configuration” on page 21-66](#).

For actions that set 802.1p bits, note that a limited set of policy conditions are supported. For information about which conditions can be used with an 802.1p action, see [“Condition and Action Combinations” on page 21-29](#).

Note. 802.1p mapping can also be set for Layer 3 traffic, which typically has the 802.1p bits set to zero.

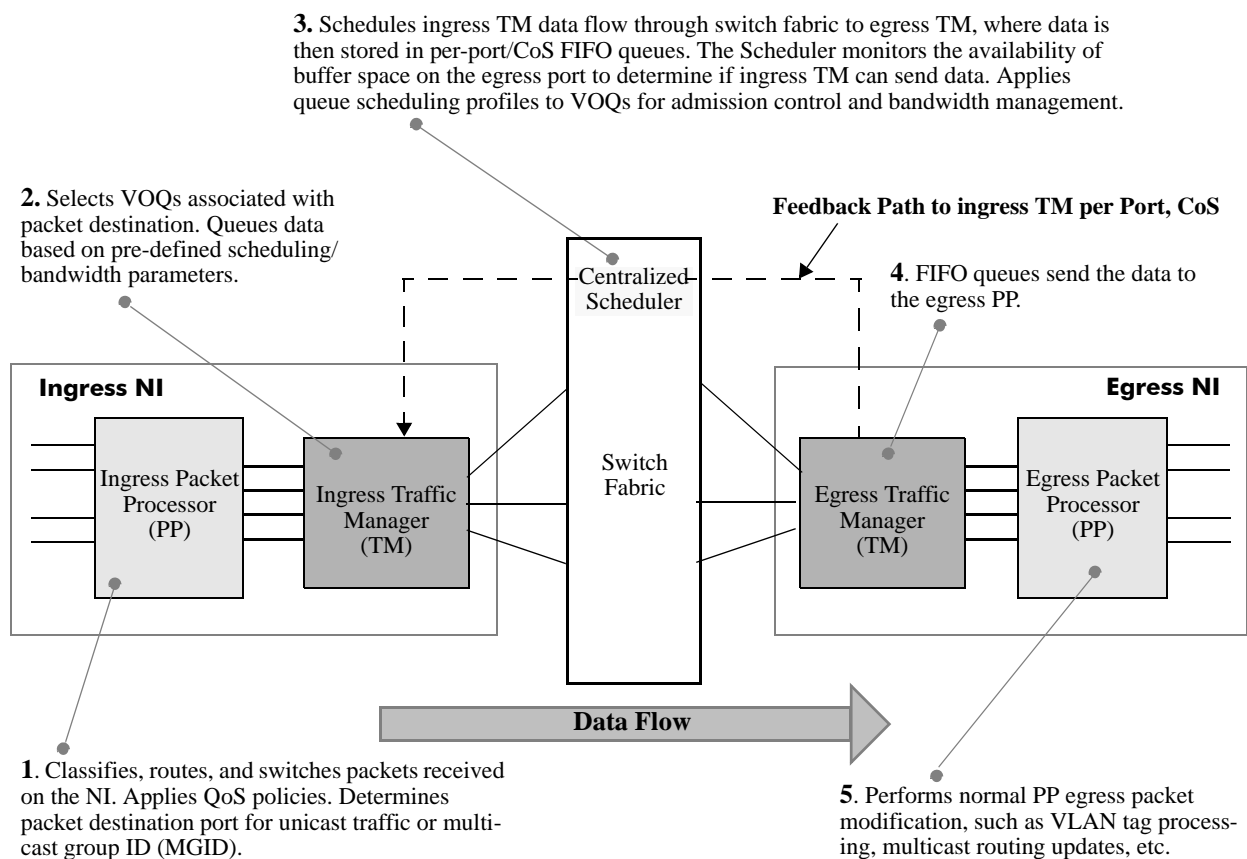
Congestion Management

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.

The OmniSwitch 10K is an input-queue switch. Queue management is done on the ingress based on a virtual output queue (VOQ) architecture that is inherent to the switch.

- The ingress queue management function on each switch NI allocates a set of eight physical VOQs for every port in the system, including ports that belong to the NI. These VOQs are “virtual” in that they serve as egress queues for a port but are located and managed on the ingress.
- Extended memory on the ingress provides deep packet buffering to absorb short periods of congestion without dropping traffic.
- An advanced feedback mechanism from egress to the ingress buffers traffic on ingress to prevent head-of-line (HOL) blocking.
- A centralized scheduler schedules traffic to every port based on inputs from all the ingress VOQs destined to the egress port.
- Traffic is scheduled through the switch fabric based on the egress port speed, thus providing efficient use of the fabric bandwidth.

The following diagram shows the components within the system that provide the QoS features and input-based queuing using VOQs:



This implementation of VOQ means that packets only cross the switch shortly before they exit the system. And, because egress bandwidth capacity is checked before packets are sent across the switch, adverse traffic patterns do not disrupt rate guarantees.

Queue Sets

The VOQ queue management and related QoS functions are implemented using a framework based on Queue Sets (QSets). A QSet is a set of aggregate VOQs that are associated with each port.

The QSET framework involves the following elements:

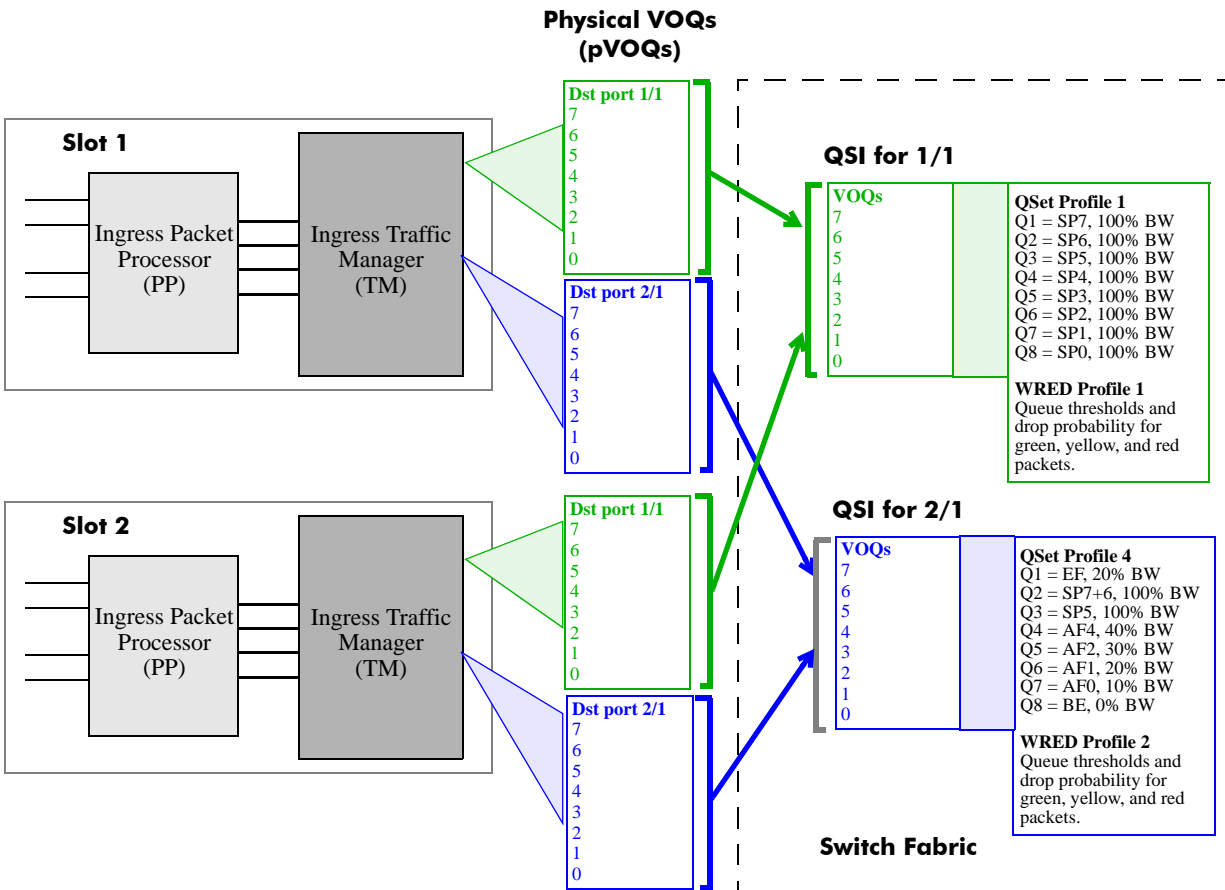
- **QSet instance (QSI)**—A QSI is a logical entity that refers to a set of eight VOQs that reside on the switch fabric. Each port in the switch is automatically associated with a QSI. The QSI represents all the ingress Traffic Manager (TM) physical VOQs (pVOQs) associated with the port to which the QSI is attached.
- **QSet profile (QSP)**—a profile associated with each QSI that defines the output scheduling behavior for the ingress TM pVOQs associated with the QSet instance. There are four pre-defined profiles, with QSP 1 serving as the default profile that is automatically assigned to each QSI. See [“QSet Profiles” on page 21-13](#).
- **Weighted Random Early Detection (WRED) profile**—a WRED profile (WRP) is associated with each QSet profile to provide admission control for each QSet. Up to four WRPs are supported, with WRP 1 serving as the default profile automatically assigned to each QSP. Profiles 2–4 are user-configurable to apply custom WRED settings. See [“WRED Profiles” on page 21-15](#).

How it Works

When a physical switch port comes up, a QSet instance (a set of eight logical VOQs) is automatically associated with the port for unicast traffic. In addition, the default QSet profile (QSP 1) is automatically assigned to the QSI, and QSP 1 imports the default WRED profile (WRP 1).

If a port attaches to a link aggregate (LAG), a QSI and default QSP 1 is 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 (QSP 1).

The following diagram shows the relationship between switch ports, QSet instances, and QSet profiles as they apply to unicast traffic. See [“Multicast VOQ” on page 21-13](#) for more information.



Queue Set (QSet) Framework (Unicast Traffic)

In this example illustration:

- Ingress packets destined for port 1/1 and port 2/1 are queued into CoS/VOQs allocated by the ingress TM. These are the physical VOQs (pVOQs), that reside on each NI for every port in the switch.
- The centralized scheduler within the switch fabric maintains a QSI for port 1/1 and port 2/1.
- The QSP assigned to the QSI for port 1/1 and port 2/1 is applied by the scheduler to all pVOQ flows destined for port 1/1 or port 2/1.
- There are four pre-defined QSPs supported. In this example, the default QSP 1 is associated with the QSI for port 1/1. However, QSP 4 was assigned to the QSI for port 2/1.
- QSP 1 applies the default WRED profile 1 (WRP 1) to the QSI for port 1/1. QSP 4 applies the user-defined WRP 4 to the QSI for port 2/1.
- The QSet framework shown in this example applies to unicast traffic. Selecting QSPs and configuring WRPs only applies to unicast VOQ queue management.

Note that the VOQs that reside in the switch fabric, as represented by QSet instances, is just a state space where configured and dynamic parameters for each VOQ is specified. There is no physical buffer memory associated with these VOQs in the fabric. Packets are not buffered in fabric

Multicast VOQ

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 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.

QSet Profiles

There are four pre-defined QSet profiles available (QSP 1 - 4). Each profile defines the following bandwidth management attributes that are applied to VOQ flows 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 name and ID and administrative status of the WRED profile (WRP) that is applied to all of the QSet queues. By default, WRP 1 is associated with each QSet profile.
- The administrative status of statistics collection for the QSet queues.
- The queue specific (QSpec) priority used for output scheduling on each of the 8 QSet queues.

The following four pre-defined QSet profiles are supported:

QSP	Default WRP	Bandwidth	Queue Specific Priority
QSP 1	WRP 1	100%	8 SP
QSP 2	WRP 1	100%	1 EF, 7 SP
QSP 3	WRP 1	100%	1 EF, 7 WFQ
QSP 4	WRP 1	100%	1 EF, 2 SP, 4 AF, 1 BE

To determine how flows are mapped to VOQs based on ingress priority markings, see [“Queue Mapping Tables” on page 21-16](#). This section contains CoS priority mapping tables for each QSet profile.

Configuring QSet Profiles

The default 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 configure or replace this profile if the default profile attributes are not sufficient.

Consider the following when configuring a QSet profile:

- QSP 1, 2, 3, and 4 cannot be deleted from the switch configuration.
- Creating new profiles is not supported; only the four QSPs provided are available for selection at this time.
- Configurable parameters for the profiles include the WRED profile (WRP) association and the administrative status for the WRP and statistics collection for the QSet instance. By default, WRED and statistics collection are enabled for the profile.

- There is only one QSet profile per 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 link aggregate, the port is associated with QSP 1 when it leaves the LAG.

The **qos qset** command is used to configure WRED and statistics parameters for QSP 1, 2, 3, or 4. The **qos qsi** command is used to select the QSP for a specific QSet instance (QSI). For example, the following commands select WRP 2, disable statistics collection, and specify QSP2 as the profile for the QSet instance associated with port 1/2:

```
-> qos qset qsp-id 2 wrp-id 2
-> qos qset qsp-id 2 wrp-id 2 wred-admin enable
-> qos qset qsp-id 2 stats-admin disable
-> qos qsi port 1/2 qsp-id 2
```

To verify the QSet profile configuration for the switch, use the **show qos qset** command. For example:

```
-> show qos qset qsp-id 2
QSP 2 (qsp-2)
#Ports: 0, #Queues: 8, BW (%): 100, Type: Default,
WRP: 1, Name: wrp-1, Admin: Ena,
Scheduler: Qspec, Stats-Admin: Ena
QP 1
Qtype: EF, Stats-Admin: Ena,
WRP: 1, Name: wrp-1, Admin: Ena,
CIR (%): 0, PIR (%): 20
QP 2
Qtype: SP7, Stats-Admin: Ena,
WRP: 1, Name: wrp-1, Admin: Ena,
CIR (%): 0, PIR (%): 100
QP 3
Qtype: SP5, Stats-Admin: Ena,
WRP: 1, Name: wrp-1, Admin: Ena,
CIR (%): 0, PIR (%): 100
QP 4
Qtype: SP4, Stats-Admin: Ena,
WRP: 1, Name: wrp-1, Admin: Ena,
CIR (%): 0, PIR (%): 100
QP 5
Qtype: SP3, Stats-Admin: Ena,
WRP: 1, Name: wrp-1, Admin: Ena,
CIR (%): 0, PIR (%): 100
QP 6
Qtype: SP2, Stats-Admin: Ena,
WRP: 1, Name: wrp-1, Admin: Ena,
CIR (%): 0, PIR (%): 100
QP 7
Qtype: SP1, Stats-Admin: Ena,
WRP: 1, Name: wrp-1, Admin: Ena,
CIR (%): 0, PIR (%): 100
QP 8
Qtype: SP0, Stats-Admin: Ena,
WRP: 1, Name: wrp-1, Admin: Ena,
CIR (%): 0, PIR (%): 100
```

To verify the QSet instance associations for a port or LAG, use the **show qos qsi** command. For example:

```
-> show qos qsi port 1/1
Port 1/1
  QSAP: 1, Parent: 1, Admin: Ena
  QSI 1
    QSP: 1, Name: qsp-1, Type: Default,
    WRP: 1, Name: wrp-1, Admin: Dis, Oper: Dis,
    Scheduler: Qspec, Stats-Admin: Dis, Stats-Oper: Dis
    BW
      Admin (%): 100, Oper (Mbps): 0
```

See the *OmniSwitch CLI Reference Guide* for more information about the **qos qset**, **qos qsi**, and related **show** commands.

WRED Profiles

Weighted Random Early Detection (WRED) is a queue management function that monitors the virtual output queues of the switch to provide early detection and notification of queue congestion. For more information about WRED and configuring WRED profiles, see [“Congestion Avoidance” on page 21-18](#).

Queue Mapping Tables

This sections contains a unicast queue mapping table for each of the four pre-defined QSet profiles.

QSet Profile 1 (Default)

Egress Queue ID	Queue Type	Scheduling Algorithm	Max BW (% of Port Rate)	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

Egress Queue ID	Queue Type	Scheduling Algorithm	Max BW (% of Port Rate)	802.1p	ToS	DSCP	Notes
1	EF	SP	20%	X(5)	X(5)	5.6	Protected EF with tunable PIR
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

Egress Queue ID	Queue Type	Scheduling Algorithm	Max BW (% of Port Rate)	802.1p	ToS	DSCP	Notes
1	EF	SP	20%	X(5)	X(5)	5.6	Protected EF with tunable limit
2	WFQ7+6	WFQ	20%	7, 6	7, 6	7.x, 6.x	WFQ with tunable CIR
3	WFQ5	WFQ	12%	5	5	5.x	WFQ with tunable CIR
4	WFQ4	WFQ	12%	4	4	4.x	WFQ with tunable CIR
5	WFQ3	WFQ	12%	3	3	3.x	WFQ with tunable CIR
6	WFQ2	WFQ	38%	2	2	2.x	WFQ with tunable CIR
7	WFQ1	WFQ	4%	1	1	1.x	WFQ with tunable CIR
8	WFQ0	WFQ	2%	0	0	0	WFQ with tunable CIR

QSet Profile 4

Egress Queue ID	Queue Type	Scheduling Algorithm	Max BW (% of Port Rate)	802.1p	ToS	DSCP	Notes
1	EF	SP	20%	X(5)	X(5)	5.6	Protected EF with tunable PIR
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%	4	4	4.1, 4.2, 4.3	AF4 WFQ with tunable CIR (starvable)
5	AF3	WFQ	30%	3	3	3.1, 3.2, 3.3	AF3 WFQ with tunable CIR (starvable)
6	AF2	WFQ	20%	2	2	2.1, 2.2, 2.3	AF2 WFQ with tunable CIR (starvable)
7	AF1	WFQ	10%	1	1	1.1, 1.2, 1.3	AF1 WFQ with tunable CIR (starvable)
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

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.

The OmniSwitch 10K uses Weighted Random Early Detection (WRED) for admission control and congestion avoidance. WRED continuously monitors the average queue length over time. When the average length exceeds specific queue thresholds, WRED begins to randomly drop packets with a specific probability.

WRED uses the drop precedence (color) of the packet to determine which packets to drop. 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 [“Configuring Tri-Color Marking” on page 21-22](#) for more information). Note that all packets that are not marked with a specific color are treated as green (committed) packets.

WRED Profiles

This implementation of WRED uses drop profiles to apply active queue management to VOQs. Each profile specifies queue thresholds (as a percentage of maximum queue size) and drop probability values for green, yellow, and red traffic. The thresholds specify a linear relationship between average queue length and drop probability.

A WRED drop profile (WRP) defines the following drop precedence parameters for packets marked green, yellow, and red:

- The minimum queue length threshold (percentage of queue size).
- The maximum queue length threshold (percentage of queue size).
- The drop probability (percentage of packets to be dropped).
- The queue gain (a numeric value, 1–15, used to determine the instantaneous average queue length).

Maximum Queue Size

As previously mentioned, the minimum and maximum queue length threshold is a function of the maximum queue size. Note that the queue size is an aggregate of all the ingress Traffic Manager queues that are allocated for a specific port.

For example, if the maximum queue size for a port is 1.5MB for each of its queues per ingress Traffic Manager (also referred to as a node) and there are 8 48-port NIs in the system, each port with a maximum queue size of 1.5MB, the maximum aggregated queue size is calculated as $1.5\text{MB} * 8 = 12\text{MB}$.

In this example, if the WRED profile applied to the port specifies a minimum and maximum queue threshold of 30 and 70 percent:

- The minimum queue length threshold is 3.6 MB (30% of 12MB).
- The maximum queue length threshold is 8.4 MB (70% of 12MB).

If any NI is removed or inserted, this causes the addition or removal of Traffic Manager queues, which in turn changes the aggregate maximum queue size. When this occurs, the system implicitly takes care of adjusting all the WRED profile minimum and maximum threshold values to reflect the new aggregate maximum queue size.

Queue Gain

The gain value is used to determine the instantaneous average queue length. The instantaneous average queue length is a function of the previous average queue length and the current aggregate queue length aggregated from all the Traffic Managers (nodes) in the system.

The higher the gain value, the more of the previous average queue length is accounted for when computing the instantaneous average queue length. A higher gain value helps the system to absorb bursts of traffic.

The lower the gain value, the more of the current queue length is accounted for when computing the instantaneous average queue length. A lower gain value is useful when there is a stream with a steady flow rate of traffic. However, if the gain value is too low, the system may not respond to congestion quickly enough to initiate the random drops.

Configuring WRED Profiles

By default, WRP 1 is associated with each QSet profile and applied to each queue within the QSet instance associated with the QSet profile. If the default profile is not sufficient, configuring up to three additional profiles (WRP 2-4) is supported.

The default WRP 1 profile applies the following queue length thresholds, drop probability, and gain parameter values:

Color	Min Queue	Max Queue	Drop Probability	Gain
Green	40	90	24	9
Yellow	30	90	30	9
Red	20	90	36	9

Consider the following guidelines when configuring WRED profiles:

- The default WRP 1 profile is not configurable and cannot be deleted from the switch configuration.
- When a new profile is created, the default WRP 1 profile settings are imported into the new profile, unless new parameters are specified at the time the profile is created.
- Administratively disabling a profile is required before any changes can be made to the profile settings.
- Configuring and applying queue threshold values is similar to configuring and applying bandwidth shaping rates to limit bandwidth.
- Configure the minimum queue threshold value large enough to account for bursty traffic. If this value is too small, packet dropping will occur and may cause under utilization of the link.

-
- The maximum queue threshold determines the delay a packet may experience while waiting for the queue to clear out other packets that arrived first. A good rule of thumb is to set this value to twice the size of the minimum queue threshold.

Configuring WRED profiles involves the following tasks:

- Creating a new WRED profile, if necessary, using the **qos wred** command. For example:

```
-> qos wred wrp-id 2 color g min 10 max 80 pb 2 gain 7 y min 20 max 80 pb 3 gain
7 r min 30 max 80 pb 4 gain 7
-> qos wred wrp-id 2 admin-state enable
```

- Changing the WRED profile associated with the QSet profile to which the WRED parameters will apply. This is done using the **qos qset** command.

```
-> qos qset qsp-id 2 wrp-id 2
```

To verify the WRED profile configuration, use the **show qos wred** command. For example:

```
-> show qos wred
WRP 1 (wrp-1)
  Admin: Ena, #Ports: 480, MTU: 1540
  Red
    Min-Th: 20, Max-Th: 90, Max-Pb: 36, Gain: 9
  Yellow
    Min-Th: 30, Max-Th: 90, Max-Pb: 30, Gain: 9
  Green
    Min-Th: 40, Max-Th: 90, Max-Pb: 24, Gain: 9
```

See the *OmniSwitch CLI Reference Guide* for more information about the **qos wred**, **qos qset**, and **show qos wred** commands.

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 10K 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 21-21](#).
- **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 21-23](#) for more information.
- **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 27, “Configuring VLAN Stacking,”](#) for more information.

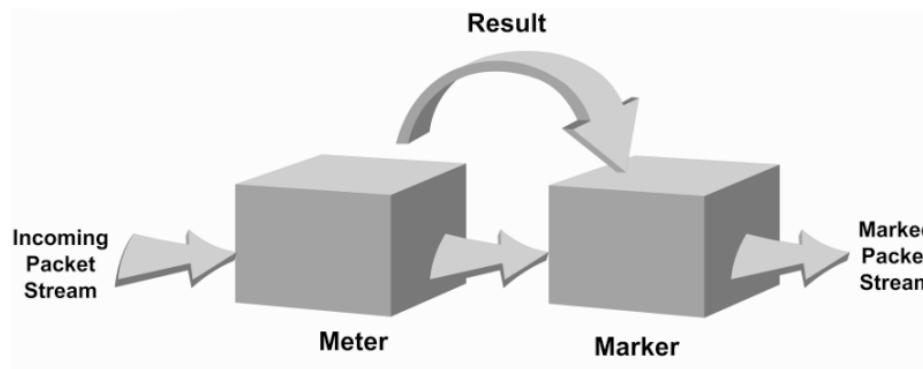
Shaping

- **Port-based QoS bandwidth shaping.** The QoS CLI provides two commands for setting the maximum ingress and egress bandwidth for a specific port. These QoS port parameters define the rate at which traffic is received and sent on the specified port. See [“Configuring Policy Bandwidth Policing” on page 21-23](#).
- **Queue bandwidth shaping.** This type of shaping is implemented through the virtual output queue architecture of the switch. A set of VOQs for each port (QSet) is associated with a profile that defines and applies the shaping and scheduling configuration for each VOQ in the QSet. See [“Congestion Management” on page 21-10](#) 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:



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 either single-rate or two-rate mode. There is no explicit

command to select the type of TCM. See [“Configuring Tri-Color Marking” on page 21-22](#) 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.
	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 21-21](#) 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.

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 21-70](#).

Port Groups and Maximum Bandwidth

If a port group condition (see “[Creating Port Groups](#)” on page 21-53) 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, 3/10, and 4/5, then the actual bandwidth limit enforced for all three ports is 30M.
- The maximum traffic received by a destination port is also dependant 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 limitation of bridged port traffic only on OmniSwitch 10K destination ports applies in this case as well.

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 1/1-2
-> 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 1/25 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 Shaping

QoS supports configuring maximum bandwidth on ingress and egress ports through the **qos port maximum egress-bandwidth** and **qos port maximum ingress-bandwidth** CLI commands. For more information about these commands, see the *OmniSwitch 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.

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. Policies 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 chapter discusses policy configuration using the CLI. For information about using WebView to configure the switch, see the *OmniSwitch AOS Release 7 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 21-45](#).
- ***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 21-10](#).

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 7, “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 22, “Managing Policy Servers.”](#)
- **VLAN Stacking Ethernet Service**—VLAN Stacking ports are always trusted and 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 27, “Configuring VLAN Stacking.”](#)

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 21-27](#) and [“Policy Actions” on page 21-28](#).

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 source VLAN group inner source VLAN inner source VLAN group destination VLAN (multicast rules only) destination VLAN group (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 next header (NH) 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.
- Source and destination MAC address conditions cannot be used in IPv6 policy rules.

-
- 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.
 - 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 21-39](#).

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 IP
 - 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 21-39.

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 <i>or</i> 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 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
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 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).

Queue Management Defaults

The OmniSwitch 10K implements a Virtual Output Queuing (VOQ) architecture that provides differentiated services (Diffserv) class of service (CoS). Based on the Diffserv recommendation, the OmniSwitch 10K includes the use of forwarding classes configured and applied through embedded profiles.

Each port and link aggregate is associated with a set of 8 VOQs, referred to as a queue set (QSet) instance. The QSet instance (QSI) is associated with QSet profile 1 (QSP 1) by default. The default QSP 1 imports a default Weighted Random Early Detection (WRED) profile (WRP 1).

A QSP defines both global parameters for the profile and individual queue profile parameters that are applied to the 8 VOQs associated with the QSet instance. See [“Congestion Management” on page 21-10](#) for more information.

The following are the default QSet and queue profile settings applied with QSP 1:

QSP 1	Default
Bandwidth	100%
WRED Profile	WRP 1
WRP 1 Admin Status	Enabled
Statistics Admin Status	Enabled
QP1–QP8 Queue Type	Strict Priority
QP1–QP8 Admin Status	Enabled
QP1–QP8 WRP	WRP 1
QP1–QP8 WRP Admin Status	Enabled
QP1–QP8 CIR PIR	0%, 100%

The following are the default drop precedence settings applied with WRP 1:

WRP 1	Green	Yellow	Red
Minimum queue threshold	40%	30%	20%
Maximum queue threshold	90%	90%	90%
Drop probability	24%	30%	36%
Gain	9	9	9

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	enabled
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 are counted
Whether to send a trap for the rule.	trap	enabled (trap sent only on port disable action or UserPort shutdown operation).

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		Single-rate TCM (srTCM) mode
- committed rate and burst size	cir cbs	CIR=0, CBS=10K
- peak rate and burst size	pir pbs	PIR=0, PBS=10K

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** network group*—The switch has a default network group, called **switch**, that includes all IP addresses configured for the switch itself. This default network group can be used in policies. See [“Creating Network Groups” on page 21-49](#) 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 21-30](#) for a list of the global defaults. See [“Configuring Global QoS Parameters” on page 21-35](#) 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 21-31](#) for a list of port defaults. See [“Classification” on page 21-6](#) and [“Traffic Policing and Shaping” on page 21-20](#) for information about configuring port parameters.

3 Configuring Queue Set (QSet) Profiles. The queue management configuration for the ingress virtual output queues (VOQ) 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 21-31](#) for a list of default profile settings. See [“Congestion Management” on page 21-10](#) for information about configuring QSet profiles.

4 Setting Up Policies. Most QoS configuration involves setting up policies. See [“Creating Policies” on page 21-39](#).

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 21-66](#).

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 21-36](#).

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 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 21-66](#).

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 21-66](#).

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 21-66](#).

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 7 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 yubal (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 yubal(1)
Enable rule yubal (2) 1,1
Really enable yubal
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 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 21-30.

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 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 21-69](#).

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 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 21-66](#).

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 21-26](#).

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 21-66](#).

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 21-48](#) 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 21-27](#).

policy condition keywords

source ip	service	source port
source ipv6	service group	source port group
destination ip	ip protocol	destination port
destination ipv6	icmp type	destination port group
source network group	icmp code	
destination network group	802.1p	ipv6
source ip port	inner 802.1p	nh
destination ip port	tos	flow-label
source tcp port	dscp	
destination tcp port		
source udp port	source mac	
destination udp port	destination mac	
established	source mac group	
tcpflags	destination mac group	
	source vlan	
	source vlan group	
	inner source vlan	
	inner source vlan group	
	destination vlan (multicast only)	
	ethertype	

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 21-43](#) 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 21-66](#).

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 21-27](#) and [“Policy Actions” on page 21-28](#). See the *OmniSwitch CLI Reference Guide* for details about command syntax.

policy action keywords

disposition	dcsp
shared	map
priority	port-disable
maximum bandwidth	redirect port
maximum depth	redirect linkagg
cir cbs pir pbs	no-cache
tos	mirror
802.1p	

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 21-43](#) 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 21-66](#).

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 21-32](#).

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 **policy validity-period** 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 21-35](#).

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 21, “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 **issu slot** 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 7 Switch Management Guide* and the *OmniSwitch CLI Reference Guide*.

For more information about applying rules, see [“Applying the Configuration” on page 21-66](#).

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**.

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 21, “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 21-59.](#)

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 21-41.](#)

```
-> 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 CLI Reference Guide* for more information about the output of this display. See [“Verifying Condition Group Configuration” on page 21-55](#) 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 21-43.](#)) 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 21-66](#) for more information about this command.

```
-> qos apply
```

The next sections describe how to create groups in more detail.

Creating Network Groups

Use network policy groups for policies based on IPv4 source or destination addresses. Note that IPv6 addresses are not supported with network groups at this time. The policy condition specifies whether the network group is a source network group, destination network group, or multicast network group.

- **Default switch group**—Note that by default the switch contains a network group called **switch** that includes all IPv4 addresses configured for the switch itself. This network group 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 21-59.](#)

To create a network policy group, use the **policy network group** command. Specify the name of the group and the IPv4 address(es) to be included in the group. Each IPv4 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 21-41](#) 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	<i>The protocol must be specified with at least one source or destination port.</i>
Shortcut for TCP service	source tcp port destination tcp port	<i>Keywords can be used in combination.</i>
Shortcut for UDP service	source udp port destination udp port	<i>Keywords can be used in combination.</i>

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 21-51](#).

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 21-50](#) 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 21-41.](#))

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 21-41](#) 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 21-41.](#)

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 21-41](#) 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 3/1 3/2 3/3
```

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 21-41](#).

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
```

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 21-41](#) 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 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 21-57](#) and [“Creating Map Groups” on page 21-57](#).

```
-> 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 21-42](#) 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 21-58](#) and the *OmniSwitch 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 21-43](#).

```
-> policy rule r3 condition Traffic action tosMap
```

4 Apply the configuration. For more information about this command, see [“Applying the Configuration” on page 21-66](#).

```
-> 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 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 21-26 and “Creating Policies” on page 21-39.

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

In this example, the default bridged disposition is **accept** (the default). Since the default is **accept**, the **qos default bridged disposition** command would only need to be entered if the disposition had previously been set to **deny**. The command is shown here for completeness.

```
-> qos default bridged disposition accept
-> policy condition Address1 source mac 080020:112233 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 1/4-6 1/8 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

In this example, the default routed disposition is **accept** (the default). Since the default is **accept**, the **qos default routed disposition** command would only need to be entered if the disposition had previously been set to **deny**. The command is shown here for completeness.

```
-> qos default routed disposition accept
-> 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 2 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. The default disposition is set to **deny**.

```
-> qos default routed disposition deny
-> 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 Layer 2 conditions, 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 ipv6	destination udp port
destination ipv6	ipv6
source tcp port	nh (next header)
destination port	flow-label
source udp port	

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.

Note the following when configuring IPv6 ACLs:

- Trusted/untrusted behavior is the same for IPv6 traffic as it is for IPv4 traffic.
- IPv6 policies do not support the use of network groups, service groups, map groups, or MAC groups.
- IPv6 multicast policies are not supported.
- Anti-spoofing and other UserPorts profiles/filters do not support IPv6.
- The default (built-in) network group, “Switch”, only applies to IPv4 interfaces. There is no such group for IPv6 interfaces.

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 20, “Configuring IP Multicast Switching.”](#)

Multicast traffic has its own global disposition. By default, the global disposition is **accept**. To change the default, use the **qos default multicast disposition** command.

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:

```
-> qos default multicast disposition deny
-> policy condition Mclient1 multicast ip 224.0.1.2 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 224.0.1.2 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 21-64](#).
- **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 **icmrcode**. See [“Configuring ICMP Drop Rules” on page 21-64](#).
- **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 21-65](#).
- **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.

By default spoofed traffic is filtered on user ports. To specify additional types of traffic to look for on these ports and select how the port deals with such traffic, use the **qos user-port** command to configure a **UserPorts** profile. For example, the following command specifies that user ports must filter BPDU packets:

```
-> qos user-port filter spoof
```

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
```

Note that a slot and port 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.

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
```

Note that 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.

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
```

Note that any changes to the **UserPorts** profile (e.g., adding or removing a traffic type) are not made until the **qos apply** command is performed.

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 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 CLI Reference Guide* for more information.

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 21-43](#).

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	rule1
rule6	rule2
	rule3

If you enter **qos revert**, the configuration then looks like:

Pending Policies	Applied Policies
rule1	rule1
rule2	rule2
rule3	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	rule1
rule6	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 Policy-View application. Separate commands are used for loading and flushing LDAP policies on the switch. See [Chapter 23, “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 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 ip
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 21, “Configuring QoS.”](#)

Policies can also be used for prioritizing traffic in dynamic link aggregation groups. For more information about dynamic link aggregates, see [Chapter 7, “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 21-48](#) 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 21-27 and “Policy Actions” on page 21-28.

Basic Commands

The following **policy action** commands are used for traffic prioritization or policing (rate limiting):

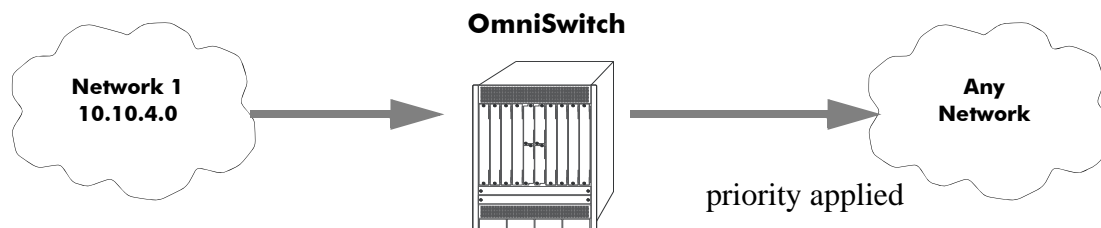
policy action priority
policy action maximum bandwidth
policy action maximum depth

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 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.



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 ceases 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 are 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 mirror ingress 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 mirror ingress 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.

Note the following regarding the use and configuration of mirroring policies:

- Only one policy-based MTP session is supported at any given time. As a result, all mirroring policies must specify the same destination port.
- 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 802.1p 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 21-8](#).

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
```

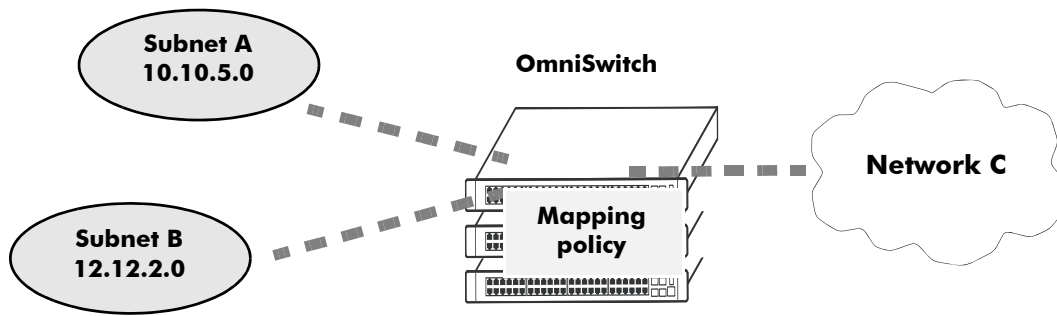
In the next example, the **policy map group** command specifies a group of values that must be mapped; the **policy action map** command specifies what must be mapped (802.1p to 802.1p, ToS/DSCP to 802.1p) and the mapping group that must be used. For more details about creating map groups, see [“Creating Map Groups” on page 21-57](#).

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 is 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
```



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.

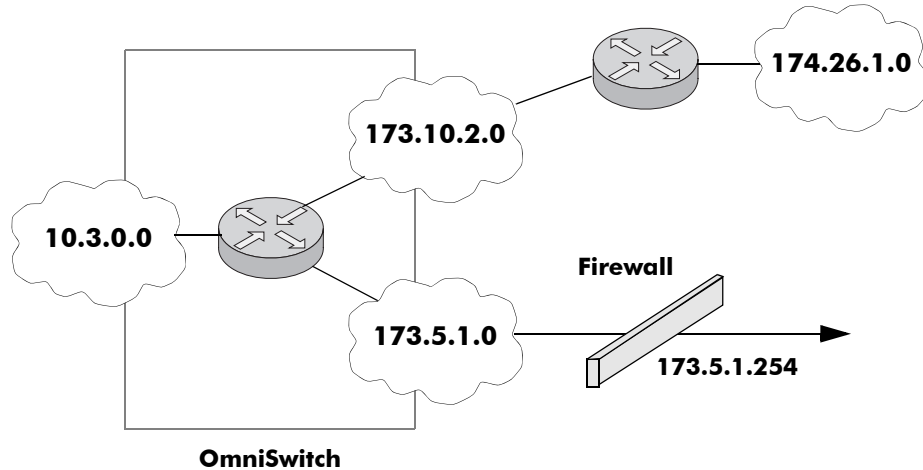
Note. When a PBR QoS rule is applied to the configuration, it is applied to the entire switch, unless you specify a built-in port group in the policy condition.

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 11, “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.



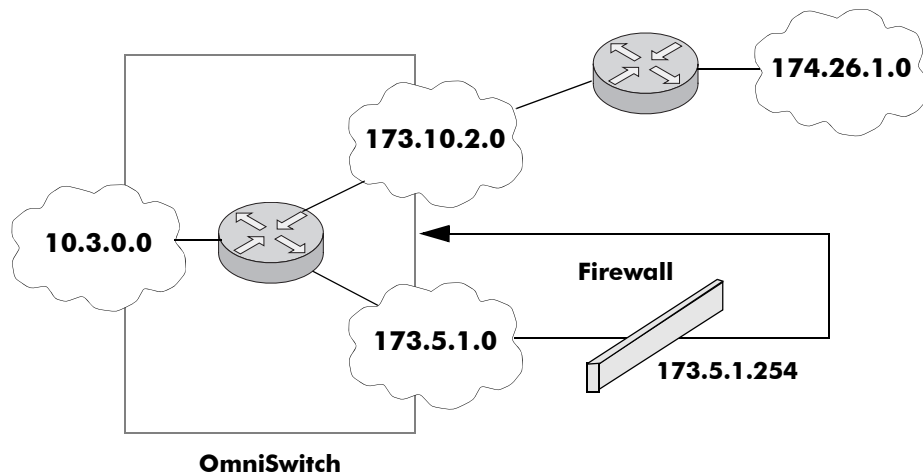
Routing all IP source traffic through a firewall

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:



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.

22 Managing Policy Servers

Quality of Service (QoS) policies that are configured through Alcatel-Lucent's 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 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 22-3](#)
- [“Modifying Policy Servers” on page 22-4](#)
- [“Verifying the Policy Server Configuration” on page 22-7](#)

Policy Server Specifications

The following table lists important information about LDAP policy servers:

Platforms Supported	OS10K
LDAP Policy Servers RFCs Supported	RFC 2251–Lightweight Directory Access Protocol (v3) RFC 3060–Policy Core Information Model—Version 1 Specification
Maximum number of policy servers (supported on the switch)	8
Maximum number of policy servers (supported by PolicyView)	1

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

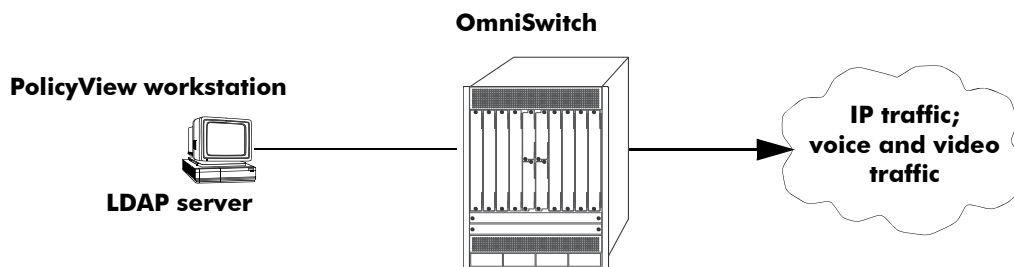
Policy Server Overview

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 21, “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.



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 22-2](#).

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 22-5](#).

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 policies are applied from PolicyView or vice versa, it activates all current configuration.

For more information about configuring policies through the CLI, see [Chapter 21, “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.

23 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 23-7.
- **Configuring a TACACS+ Server.** This procedure is described in [“TACACS+ Server”](#) on page 23-12.
- **Configuring an LDAP Server.** This procedure is described in [“LDAP Servers”](#) on page 23-14.

For information about using servers for authenticating users to manage the switch, see the “Switch Security” chapter in the *OmniSwitch AOS Release 7 Switch Management Guide*.

Authentication Server Specifications

Platforms Supported	OS10K
RADIUS RFCs Supported	RFC 2865–Remote Authentication Dial In User Service (RADIUS) RFC 2866–RADIUS Accounting RFC 2867–RADIUS Accounting Modifications for Tunnel Protocol Support RFC 2868–RADIUS Attributes for Tunnel Protocol Support RFC 2809–Implementation of L2TP Compulsory Tunneling through RADIUS RFC 2869–RADIUS Extensions RFC 2548–Microsoft Vendor-specific RADIUS Attributes RFC 2882–Network Access Servers Requirements: Extended RADIUS Practices
TACACS+ RFCs Supported	RFC 1492–An Access Control Protocol
LDAP RFCs Supported	RFC 1789–Connectionless Lightweight X.5000 Directory Access Protocol RFC 2247–Using Domains in LDAP/X.500 Distinguished Names RFC 2251–Lightweight Directory Access Protocol (v3) RFC 2252–Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions RFC 2253–Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names RFC 2254–The String Representation of LDAP Search Filters RFC 2256–A Summary of the X.500(96) User Schema for Use with LDAPv3
Other RFCs	RFC 2574–User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) RFC 2924–Accounting Attributes and Record Formats RFC 2975–Introduction to Accounting Management RFC 2989–Criteria for Evaluating AAA Protocols for Network Access
Maximum number of authentication servers in single authority mode	8
Maximum number of authentication servers in multiple authority mode	8
Maximum number of servers per Authenticated Switch Access type	8

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 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
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 23-7, “TACACS+ Server” on page 23-12, and “LDAP Servers” on page 23-14.

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 *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 7 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 (Authenticated VLANs).

RADIUS, TACACS +, or LDAP servers can be used for Authenticated Switch Access and/or Authenticated VLANs. Another type of server, SecurID ACE/Server, can be used for authenticated switch access only; the ACE/Server is an authentication-only server (no authorization or accounting). Only RADIUS servers are supported for 802.1X Port-based Network Access Control.

The following table describes how each type of server can be used with the switch:

Server Type	Authenticated Switch Access	Authenticated VLANs	802.1X Port-Based Network Access Control
ACE/Server	yes (except SNMP)	no	no
RADIUS	yes (except SNMP)	yes	yes
TACACS+	yes (including SNMP)	yes	no
LDAP	yes (including SNMP)	yes	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).

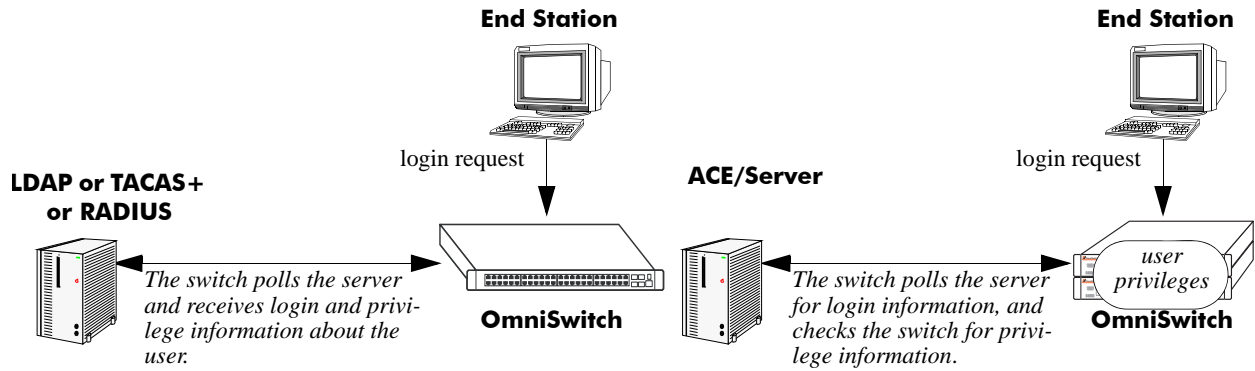
The switch uses the first available authentication server to attempt to authenticate users. If user information is not found on the first available server, the authentication attempts fails.

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.

For RADIUS, TACACS+, and LDAP, additional servers can be configured as backups.

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.



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 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 and RFC 2139, 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 RADIUS server attributes 1–39 and 60–63, their descriptions, and whether the Alcatel-Lucent RADIUS client in the switch supports them. Attribute 26 is for vendor-specific information and is discussed in [“Vendor-Specific Attributes for RADIUS” on page 23-9](#). Attributes 40–59 are used for RADIUS accounting servers and are listed in [“RADIUS Accounting Server Attributes” on page 23-10](#).

Num.	Standard Attribute	Notes
1	User-Name	Used in access-request and account-request packets.
2	User-Password	—
3	CHAP-Password	<i>Not supported.</i>
4	NAS-IP-Address	Sent with every access-request. Specifies which switches a user can have access to. More than one of these attributes is allowed per user.
5	NAS-Port	Virtual port number sent with access-request and account-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

Num.	Standard Attribute	Notes
7	Framed-Protocol	<i>Not supported. These attributes are used for dial-up sessions; not applicable to the RADIUS client in the switch.</i>
8	Framed-IP-Address	
9	Framed-IP-Netmask	
10	Framed-Routing	
11	Filter-Id	
12	Framed-MTU	
13	Framed-Compression	
14	Login-IP-Host	—
15	Login-Service	
16	Login-TCP-Port	
17	Unassigned	
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.
19	Callback-Number	<i>Not supported. These attributes are used for dial-up sessions; not applicable to the RADIUS client in the switch.</i>
20	Callback-Id	
21	Unassigned	
22	Frame-Route	
23	Framed-IPX-Network	
24	State	Sent in challenge/response packets.
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 23-9.
27	Session-Timeout	<i>Not supported.</i>
28	Idle-Timeout	<i>Not supported.</i>
29	Termination-Action	<i>Not supported. These attributes are used for dial-up sessions; not applicable to the RADIUS client in the switch.</i>
30	Called-Station-Id	
31	Calling-Station-Id	
32	NAS-Identifier	
33	Proxy-State	
34	Login-LAT-Service	
35	Login-LAT-Node	
36	Login-LAT-Group	
37	Framed-AppleTalk-Link	
38	Framed-AppleTalk-Network	
39	Framed-AppleTalk-Zone	
60	CHAP-Challenge	
61	NAS-Port-Type	
62	Port-Limit	
63	Login-LAT-Port	

Vendor-Specific Attributes for RADIUS

The Alcatel-Lucent 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, 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's vendor ID is 800 (SMI Network Management Private Enterprise Code).

The following are VSAs for RADIUS servers:

Num.	RADIUS VSA	Type	Description
1	Alcatel-Lucent-Auth-Group	integer	The authenticated VLAN number. The only protocol associated with this attribute is Ethernet II. If other protocols are required, use the protocol attribute instead.
2	Alcatel-Lucent-Slot-Port	string	Slot(s)/port(s) valid for the user.
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.
5	Alcatel-Lucent-Group-Desc	string	Description of the authenticated VLAN.
6	Alcatel-Lucent-Port-Desc	string	Description of the port.
8	Alcatel-Lucent-Auth-Group-Protocol	string	The protocol associated with the VLAN. Must be configured for access to other protocols. Values include: IP_E2 , IP_SNAP , IPX_E2 , IPX_NOV , IPX_LLC , IPX_SNAP .
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.
42	Alcatel-Lucent-Acce-Priv-F-W2	hex.	Configures functional write privileges for the user.

The Alcatel-Lucent-Auth-Group attribute is used for Ethernet II only. If a different protocol, or more than one protocol is required, use the Alcatel-Lucent-Auth-Group-Protocol attribute instead. For example:

```
Alcatel-Lucent-Auth-Group-Protocol 23: IP_E2 IP_SNAP
Alcatel-Lucent-Auth-Group-Protocol 24: IPX_E2
```

In this example, authenticated users on VLAN 23 can use Ethernet II or SNAP encapsulation. Authenticated users on VLAN 24 can use IPX with Ethernet II.

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. Specifies which switches a user can have access to. More than one of these attributes is allowed per user.
5	NAS-Port	Virtual port number sent with access-request and account-request packets. Slot/port information is supplied in attribute 26 (vendor-specific).
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.
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.
42	Acct-Input-Octets	(Authenticated VLANs only) Tracked per port.
43	Acct-Output-Octets	(Authenticated VLANs only) Tracked per port.
44	Acct-Session	Unique accounting ID. (For authenticated VLAN users, Alcatel-Lucent uses the MAC address of the client.)
45	Acct-Authentic	Indicates how the client is authenticated; standard values (1–3) are not used. Vendor specific values must be used instead: AUTH-AVCLIENT (4) AUTH-TELNET (5) AUTH-HTTP (6) AUTH-NONE (0)
46	Acct-Session	The start and stop time for a user session can be determined from the accounting log.
47	Acct-Input-Packets	(Authenticated VLANs only) Tracked per port.
48	Acct-Output-Packets	(Authenticated VLANs only) Tracked per port.
49	Acct-Terminal-Cause	Indicates how the session was terminated: NAS-ERROR USER-ERROR LOST CARRIER USER-REQUEST STATUS-FAIL

The following table lists the VSAs supported for RADIUS accounting servers. The attributes in the **radius.ini** file can be modified if necessary.

Num.	Accounting VSA	Type	Description
1	Alcatel-Lucent-Auth-Group	integer	The authenticated VLAN number. The only protocol associated with this attribute is Ethernet II. If other protocols are required, use the protocol attribute instead.
2	Alcatel-Lucent-Slot-Port	string	Slot(s)/port(s) valid for the user.
4	Alcatel-Lucent-Client-IP-Addr	dotted decimal	The IP address used for Telnet only.
5	Alcatel-Lucent-Group-Desc	string	Description of the authenticated VLAN.

Configuring the RADIUS Client

Use the **aaa radius-server** command to configure RADIUS parameters on the switch.

RADIUS server keywords

key	timeout
host	auth-port
retransmit	acct-port

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** 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
```

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 23-3](#).

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.

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.

Configuring the TACACS+ Client

Use the **aaa tacacs+-server** command to configure TACACS+ parameters on the switch.

TACACS+ server keywords

key	timeout
host	port

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** keyword).

In this example, the server name is **tacl**, 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 tacl host 10.10.5.2 10.10.5.5 key otna
```

To modify a TACACS+ server, enter the server name and the desired parameter to be modified.

```
-> aaa tacacs+-server tacl key tnmelc
```

If you are modifying the server and have just entered the **aaa tacacs+-server** command to create or modify the server, you can use command prefix recognition. For example:

```
-> aaa tacacs+-server tacl timeout 5
```

For information about server defaults, see [“Server Defaults” on page 23-3](#).

To remove a TACACS+ server, use the **no** form of the command:

```
-> no aaa tacacs+-server tacl
```

Note that only one server can be deleted at a time.

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 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 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 23-25](#) 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 attributes>	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.

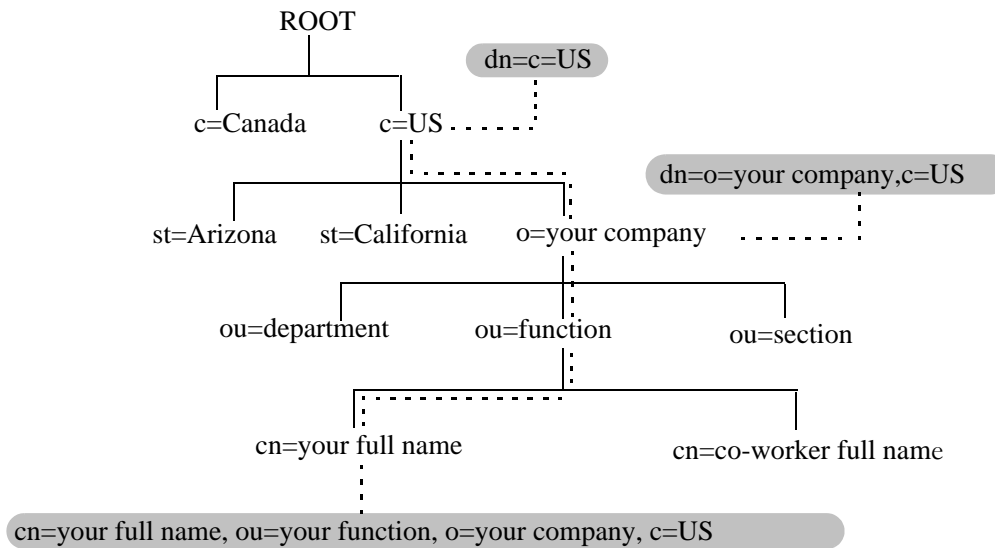
Entries are usually based on physical locations and established policies in a Directory Information Tree (DIT); the DN locates an entry in the hierarchy of the tree. Alias entries pointing to other entries can also be used to circumvent the hierarchy during searches for entries.

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.



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>	<p>Different results are retrieved depending on the scopes associated with entry searches.</p> <p>“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.</p> <p>“one” (one-level) search: retrieves information about entries one level under distinguished name (<base_dn> as specified in the URL, excluding the base entry.</p> <p>“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.</p>
<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 7 Switch Management Guide*.

Configuring Authentication Key Attributes

The alp2key tool is provided on the Alcatel-Lucent 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 “\n”) 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 “\n”) 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 “\n”)—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 <i>x</i> —for an authenticated user session, where <i>x</i> is the number of the session AVLAN —for Authenticated VLAN session in single authority mode AVLAN <i>y</i> —for Authenticated VLAN session in multiple authority mode, where <i>y</i> 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

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.

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.

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 23-4](#). For more information about the output of this command, see the *OmniSwitch CLI Reference Guide*.

24 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 24-3 or [“Deleting a Port Mapping Session”](#) on page 24-3.
- [Enabling/Disabling a Port Mapping Session](#)—see [“Enabling a Port Mapping Session”](#) on page 24-4 or [“Disabling a Port Mapping Session”](#) on page 24-4.
- [Configuring a Port Mapping Direction](#)—see [“Configuring Unidirectional Port Mapping”](#) on page 24-4 and [“Restoring Bidirectional Port Mapping”](#) on page 24-4.
- [Configuring an example Port Mapping Session](#)—see [“Sample Port Mapping Configuration”](#) on page 24-5.
- [Verifying a Port Mapping Session](#)—see [“Verifying the Port Mapping Configuration”](#) on page 24-6.

Port Mapping Specifications

Platforms Supported	OmniSwitch 10K
Ports Supported	Ethernet (10 Mbps) Fast Ethernet (100 Mbps) Gigabit Ethernet (1000 Mbps) 10 Gigabit Ethernet (10000 Mbps)
Port Mapping Sessions	16

Port Mapping Defaults

The following table shows port mapping default values.

Parameter Description	CLI Command	Default Value/Comments
Mapping Session Creation	<code>port-mapping user-port network-port</code>	No mapping sessions
Mapping Status configuration	<code>port-mapping</code>	Disabled
Port Mapping Direction	<code>port-mapping [unidirectional bidirectional]</code>	Bidirectional
Port Mapping Unknown Unicast Flooding	<code>port-mapping unknown-unicast-flooding</code>	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

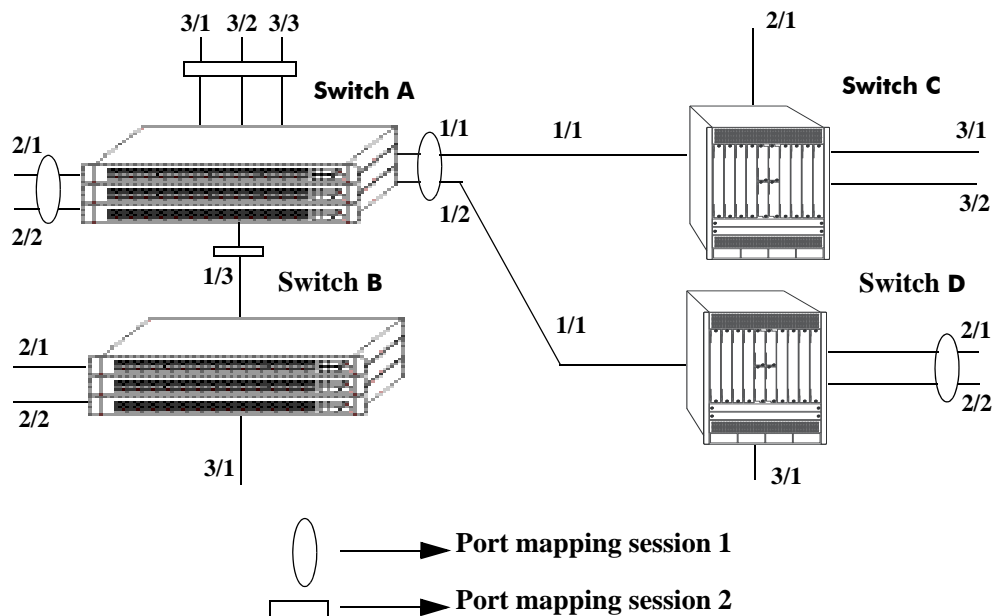
This section provides an example port mapping network configuration. In addition, a tutorial is also included that provides steps on how to configure the example port mapping session using the Command Line Interface (CLI).

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.



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 24-5](#).

- 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 CLI Reference Guide*.

25 Configuring Learned Port Security

Learned Port Security (LPS) provides a mechanism for authorizing source learning of MAC addresses on Ethernet ports. The only types of Ethernet ports that LPS does not support are link aggregate and 802.1Q trunked link aggregate 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 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.
- Two methods for handling unauthorized traffic: stopping all traffic on the port or only blocking traffic that violates LPS criteria.

In This Chapter

This chapter 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 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Enabling LPS for a port on [page 25-9](#).
- Specifying a source learning time limit for all LPS ports on [page 25-10](#).
- Configuring the maximum number of MAC addresses learned per port on [page 25-11](#).
- Configuring the maximum number of filtered MAC addresses learned per port on [page 25-11](#).
- Configuring a list of authorized MAC addresses for an LPS port on [page 25-12](#).
- Configuring a range of authorized MAC addresses for an LPS port on [page 25-12](#).
- Selecting the security violation mode for an LPS port on [page 25-12](#).
- Displaying LPS configuration information on [page 25-13](#).

For more information about source MAC address learning, see [Chapter 3, “Managing Source Learning.”](#)

Learned Port Security Specifications

Platforms Supported	OmniSwitch 10K
Ports eligible for Learned Port Security	Fixed and 802.1Q tagged
Ports not eligible for Learned Port Security	Link aggregate ports. 802.1Q (trunked) link aggregate ports. ERP ports.
Minimum number of learned MAC addresses allowed per port	100
Maximum number of filtered MAC addresses allowed per port	100
Maximum number of configurable MAC address ranges per LPS port	1

Learned Port Security Defaults

Parameter Description	Command	Default
LPS status for a port.	port-security	disabled
Number of learned MAC addresses allowed on an LPS port.	port-security port 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	disabled
MAC address range per LPS port.	port-security port 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 port 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.

1 Enable LPS on ports 6 through 8 on slot 1 using the following commands:

```
-> port-security port 1/6-8 learning-enable
```

2 Set the total number of learned MAC addresses allowed on the same ports to 25 using the following command:

```
-> port-security port 1/6-8 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 1/6-8 violation shutdown
```

Note. *Optional.* To verify LPS port configurations, use the command **show port-security**. For example:

```
-> show port-security
```

```
Port: 1/6
```

```
Operation Mode      :          SHUTDOWN-LW,
Max MAC bridged     :                10,
Violation           :          RESTRICT,
Max MAC filtered    :                5,
Low MAC Range       :    00:00:00:00:00:00,
High MAC Range      :    ff:ff:ff:ff:ff:ff
```

MAC	VLAN	MAC TYPE	OPERATION
00:00:39:59:f1:0c	1	static	bridging

```
Port: 1/7
```

```
Operation Mode      :          SHUTDOWN-LW,
Max MAC bridged     :                10,
Violation           :          RESTRICT,
Max MAC filtered    :                5,
Low MAC Range       :    00:00:00:00:00:00,
High MAC Range      :    ff:ff:ff:ff:ff:ff
```

MAC	VLAN	MAC TYPE	OPERATION

```
Port: 1/8
```

```
Operation Mode      :          SHUTDOWN-LW,
Max MAC bridged    :              10,
Violation          :          RESTRICT,
Max MAC filtered   :              5,
Low MAC Range      :      00:00:00:00:00:00,
High MAC Range     :      ff:ff:ff:ff:ff:ff
```

MAC	VLAN	MAC TYPE	OPERATION
-----+-----+-----+-----			

To verify the new source learning time limit value, use the [show port-security learning-window](#) command. For example:

```
-> show port-security learning-window
```

```
LPS Shutdown Config      = 30 min,
Convert-to-static        = DISABLE,
Remaining Learning Window = 1797 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 which the switch allows source learning to occur on all LPS ports.
- A maximum number of learned MAC addresses allowed on the port.
- A list of configured 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 two options are available for this purpose:

- Block traffic that violates LPS port restrictions; authorized traffic is forwarded on the port.
- Disable the LPS port when unauthorized traffic is received; all traffic is stopped and a port reset is required to return the port to normal operation.

LPS functionality is supported on the following Ethernet port types:

- Fixed
- 802.1Q tagged

The following port types are not supported:

- Link aggregate
- Tagged (trunked) link aggregate
- Mirrored (MTP)

How LPS Authorizes Source MAC Addresses

When a packet is received on a port that has LPS enabled, switch software checks the following criteria to determine if the source MAC address contained in the packet is allowed on the port:

- Is the source learning time window open?
- Is the number of MAC addresses learned on the port below the maximum number allowed?
- Is there a configured authorized MAC address entry for the LPS port that matches the source MAC address of the packet?

Using the above criteria, the following table shows the conditions under which a MAC address is learned or blocked on an LPS port:

Time Limit	Max Number	Configured MAC	Result
Open	Below	No entry	No LPS violation; MAC learned
Closed	Below	No entry	LPS violation; MAC blocked
Open	Above	No entry	LPS violation; MAC blocked
Open	Below	Yes; entry matches	No LPS violation; MAC learned
Closed	Below	Yes; entry matches	No LPS violation; MAC learned
Open	Above	Yes; entry matches	LPS violation; MAC blocked
Open	Below	Yes; entry doesn't match	No LPS violation; MAC learned
Closed	Below	Yes; entry doesn't match	LPS violation; MAC blocked
Open	Above	Yes; entry doesn't match	LPS violation; MAC blocked

When a source MAC address violates any of the LPS conditions, the address is considered unauthorized. The LPS violation mode determines if the unauthorized MAC address is blocked (filtered) on the port or if the entire port is disabled (see [“Selecting the Security Violation Mode” on page 25-12](#)). Regardless of which mode is selected, a notice is sent to the switch log to indicate that a violation has occurred.

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.

When a dynamically configured MAC address is added to the LPS table, it does not become a configured MAC address entry in the LPS table until the switch configuration file is saved and the switch is rebooted. If a reboot occurs before this is done, all dynamically learned MAC addresses in the LPS table are cleared.

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 2/1 and the maximum number allowed on port 2/1 is 10, then only 8 dynamically learned MAC addresses are allowed on this port.

There are two ways to define a static source MAC address entry in the LPS table; specify an individual MAC address or a range of MAC addresses. See [“Configuring an Authorized MAC Address Range” on page 25-12](#) for more information.

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 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 CLI Reference Guide* for more information about this command.

Port and Link Aggregate Security Violation

A port-security violation is triggered when the switch exceeds the limit for the maximum number of MAC addresses learnt. There are different types of violations:

- LPS violations
- QoS Policy violations
- Network Security violations
- UDLD violations
- Fabric stability related violations

A security violation occurs under the following conditions:

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

Security violations on Link Aggregates:

- When a violation occurs on a physical port that is part of a link aggregate, it 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 whole link aggregate group using the **clear violation** command, the whole link aggregate group is reactivated.
- When a simulated down violation is listed, toggling the link clears the violation, for both the link aggregates and physical ports.

A specific action is taken when a violation is detected on the port. Depending on the type of violation, two types of actions are associated with the shutdown of a port:

- **admin down** - deactivates the physical port. This action is taken for a UDLD violation.
- **simulated down** - the physical port shows as active but the applications are not allowed to access the port link. The port is put in blocking state.

The LPS violations on individual ports or link aggregates can be viewed using the **show violation** command.

Note.

- The source learning time limit is configured on the LPS ports, using the **port-security learning-window** command.
 - Use the **clear violation** command to clear all the MAC address violation logs for a particular port and session and activate the ports.
-

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 25-3](#) for a brief tutorial on configuring LPS.

Configuring LPS involves the following procedures:

- Enabling LPS for one or more switch ports. This procedure is described in [“Enabling/Disabling Learned Port Security” on page 25-9](#).
- Configuring the source learning time window during which MAC addresses are learned. This procedure is described in [“Configuring a Source Learning Time Limit” on page 25-10](#).
- 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 25-11](#).
- 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 25-11](#)
- Configuring one or more static authorized MAC addresses. This procedure is described in [“Configuring an Authorized MAC Address Range” on page 25-12](#).
- 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 25-12](#).

Enabling/Disabling Learned Port Security

By default, LPS is disabled on all switch ports. To enable LPS on a port, use the **port-security** command. For example, the following command enables LPS on port 1 of slot 4:

```
-> port-security port 4/1 learning-enable
```

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

```
-> port-security port 4/1-5 learning-enable  
-> port-security port 5/12-20 learning-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.

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

```
-> port-security 5/21-24 learning-disable
```

To disable all the LPS ports on a chassis, use the **port-security chassis learning-disable** command, as shown:

```
-> port-security chassis learning-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.

Use the **no** form of this command to remove LPS and clear all entries (configured and dynamic) in the LPS table for the specified port. For example:

```
-> no port-security port 5/10
```

After LPS is removed, all the dynamic and static MAC addresses are flushed and the learning of new MAC addresses is enabled.

Configuring a Source Learning Time Limit

By default, the source learning time limit is disabled. Use the **port-security learning-window** command to set the number of minutes the source learning window is to remain open for LPS ports. While this 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 source learning timer:

- The **port-security learning-window** command. Each time this command is issued, the timer restarts even if a current window is still open.
- Switch reboot with a **port-security learning-window** command entry saved in the **boot.cfg** file.

The LPS source learning 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
```

Once the time limit value expires, source learning of any new dynamic MAC addresses is stopped on all LPS ports even if the number of addresses learned does not exceed the maximum allowed.

Note. The LPS source learning time window has a higher priority over the maximum number of MAC addresses allowed. Therefore, if the learning interval expires before the port has learned the maximum MAC addresses allowed, the port does *not* learn anymore MAC addresses.

When the source learning time window expires, all the dynamic MAC addresses learned on the LPS ports start to age out. To prevent this, all dynamic MAC addresses must be converted to static MAC addresses. The **convert-to-static** parameter used with the **port-security learning-window** command enables or disables the conversion of dynamic MAC addresses to static MAC addresses on LPS ports when the source learning time window expires.

To enable the conversion of dynamic MAC addresses to static MAC addresses on LPS ports when the source learning time window expires, use the **port-security learning-window** command with the **convert-to-static** parameter, as shown:

```
-> port-security learning-window 30 convert-to-static enable
```

To disable the conversion of dynamic MAC addresses to static MAC addresses when the source learning time window expires, use the **port-security learning-window** command with the **convert-to-static** parameter, as shown:

```
-> port-security learning-window 30 convert-to-static disable
```

Note. The number of converted static MAC addresses cannot exceed the maximum number of MAC addresses allowed on the LPS ports.

Configuring the Number of Bridged MAC Addresses Allowed

To configure the number of bridged MAC addresses allowed on an LPS port, enter **port-security** followed by the **port** keyword and *slot/port* designation, then **maximum** followed by a number between 1 and 100. For example, the following command sets the maximum number of MAC addresses learned on port 10 of slot 6 to 75:

```
-> port-security port 6/10 maximum 75
```

To specify a maximum number of MAC addresses allowed for multiple ports, specify a range of ports. For example:

```
-> port-security port 1/10-15 maximum 10
-> port-security port 2/1-5 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 MAC addresses is stopped on the LPS port.

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 five bridged MAC addresses are learned on an LPS port, the switch sends a trap. To change the trap threshold value, use the **port-security port 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 include identifying information about the MAC, such as the address itself, the corresponding IP address, switch identification, and the slot and 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, enter **port-security** followed by the **port** keyword and *slot/port* designation, then **max-filtering** followed by a number between 1 and 100. For example, the following command sets the maximum number of filtered MAC addresses learned on port 9 of slot 5 to 18:

```
-> port-security port 5/9 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 5/9-15 max-filtering 10
-> port-security port 1/1-5 max-filtering 25
```

If the maximum number of filtered MAC addresses allowed is reached, either the LPS port is disabled (Shutdown Violation mode) or MAC address learning is disabled (Restrict Violation mode). Under both these modes, SNMP traps are generated and the events are logged in the switch log. For information on configuring the security violation modes, see [“Selecting the Security Violation Mode” on page 25-12](#).

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 source learning time limit and restrictions on the maximum number of MAC addresses allowed for the port.

To configure a source MAC address range for an LPS port, enter **port-security** followed by the **port** keyword and *slot/port* designation, **mac-range** followed by **low** and a MAC address, **high** and a MAC address. For example, the following command configures a MAC address range for port 1 on slot 4:

```
-> port-security port 4/1 mac-range low 00:20:da:00:00:10 high 00:20:da:00:00:50
```

To configure a source MAC address range specify a single port number or a range of port numbers. For example:

```
-> port-security port 4/1-5 mac-range low 00:20:da:00:00:10 high  
00:20:da:00:00:50  
-> port-security port 2/1-4 mac-range low 00:20:d0:59:0c:9a high  
00:20:d0:59:0c:9f
```

To restore the range to the default values, enter the **port-security** parameter followed by the **port** keyword and *slot/port* designation of the port and the **mac-range**. The MAC address range is restored to 00:00:00:00:00:00 and ff:ff:ff:ff:ff:ff when the **low** and **high** MAC addresses are excluded. For example, the following command sets the authorized MAC address range to the default values for port 12 of slot 4:

```
-> port-security port 4/12 mac-range
```

In addition, specifying a low end MAC and a high end MAC is optional. If either one is not specified, the default value is used. For example, the following commands set the authorized MAC address range on the specified ports to 00:da:25:59:0c:10–ff:ff:ff:ff:ff:ff and 00:00:00:00:00:00–00:da:25:00:00:9a:

```
-> port-security port 2/8 mac-range low pp:da:25:59:0c  
-> port-security port 2/10 mac-range high 00:da:25:00:00:9a
```

Refer to the *OmniSwitch CLI Reference Guide* for more information about this command.

Selecting the Security Violation Mode

By default, the security violation mode for an LPS port is set to **restrict**. In this mode, when an unauthorized MAC address is received on an LPS port, the packet containing the address is blocked. However, all other packets that contain an authorized source MAC address are allowed to forward on the port.

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.

The other violation mode option is **shutdown**. In this mode, the LPS port is disabled when an unauthorized MAC address is received; traffic is stopped at the port. The traffic is not forwarded. When a shutdown occurs, a manual reset is required to return the port back to normal operation.

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** or **shutdown**. For example, the following command selects the shutdown mode for port 1 on slot 4:

```
-> port-security port 4/1 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 4/1-10 violation shutdown
-> port-security port 1/10-15 violation restrict
```

Note. To verify the details about LPS violations, use the **show violation** command.

```
-> 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 **clear violation** command.

Displaying Learned Port Security Information

To display LPS port and table information, use the show commands listed below:

port-security port learn-trap-threshold

Displays Learned Port Security (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 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 25-3.

26 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 26-18 or [“Deleting A Mirroring Session”](#) on page 26-21.
- [Protection from Spanning Tree changes \(Port Mirroring\)](#)—see [“Unblocking Ports \(Protection from Spanning Tree\)”](#) on page 26-19.
- [Enabling or Disabling Port Mirroring Status](#)—see [“Enabling or Disabling Mirroring Status”](#) on page 26-19 or [“Disabling a Mirroring Session \(Disabling Mirroring Status\)”](#) on page 26-19.
- [Configuring Port Mirroring Direction](#)—see [“Configuring Port Mirroring Direction”](#) on page 26-20.
- [Enabling or Disabling a Port Mirroring Session](#)—see [“Enabling or Disabling a Port Mirroring Session \(Shorthand\)”](#) on page 26-20.
- [Configuring a Port Monitoring Session](#)—see [“Configuring a Port Monitoring Session”](#) on page 26-24.
- [Enabling a Port Monitoring Session](#)—see [“Enabling a Port Monitoring Session”](#) on page 26-24.

-
- [Disabling a Port Monitoring Session](#)—see “[Disabling a Port Monitoring Session](#)” on page 26-25.
 - [Deleting a Port Monitoring Session](#)—see “[Deleting a Port Monitoring Session](#)” on page 26-25.
 - [Pausing a Port Monitoring Session](#)—see “[Pausing a Port Monitoring Session](#)” on page 26-25.
 - [Configuring the persistence of a Port Monitoring Session](#)—see “[Configuring Port Monitoring Session Persistence](#)” on page 26-25.
 - [Configuring a Port Monitoring data file](#)—see “[Configuring a Port Monitoring Data File](#)” on page 26-25.
 - [Configuring a Port Monitoring direction](#)—see “[Configuring Port Monitoring Direction](#)” on page 26-26.
 - [Configuring capture-type](#)—see “[Configuring capture-type](#)” on page 26-27
 - [Displaying Port Monitoring Status and Data](#)—see “[Displaying Port Monitoring Status and Data](#)” on page 26-27.
 - [Configuring a sFlow Session](#)—see “[Configuring a sFlow Session](#)” on page 26-30.
 - [Configuring a Fixed Primary Address](#)—see “[Configuring a Fixed Primary Address](#)” on page 26-31.
 - [Displaying a sFlow Receiver](#)—see “[Displaying a sFlow Receiver](#)” on page 26-31.
 - [Displaying a sFlow Sampler](#)—see “[Displaying a sFlow Sampler](#)” on page 26-32.
 - [Displaying a sFlow Poller](#)—see “[Displaying a sFlow Poller](#)” on page 26-32.
 - [Displaying a sFlow Agent](#)—see “[Displaying a sFlow Agent](#)” on page 26-33.
 - [Deleting a sFlow Session](#)—see “[Deleting a sFlow Session](#)” on page 26-33.
 - [Enabling or Disabling RMON Probes](#)—see “[Enabling or Disabling RMON Probes](#)” on page 26-36.
 - [Configuring Resource Threshold Limits \(Switch Health\)](#)—see “[Configuring Resource Thresholds](#)” on page 26-43.
 - [Configuring Sampling Intervals](#)—see “[Configuring Sampling Intervals](#)” on page 26-45.

For information about additional Diagnostics features such as Switch Logging and System Debugging/Memory Management commands, see [Chapter 28, “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 26-14](#).

Port Mirroring Specifications

Platforms Supported	OS10K
Ports Supported	Ethernet (10 Mbps) Fast Ethernet (100 Mbps) Gigabit Ethernet (1000 Mbps) 10 Gigabit Ethernet (10000 Mbps)
Maximum Combined Local and Remote Mirroring Sessions Supported	2
N-to-1 Mirroring Supported	128 to 1
Number of RPMIR VLANs per session	1

Port Mirroring Defaults

The following table shows port mirroring default values.

Global Port Mirroring Defaults

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*) slot/ports, 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 2/3-9 destination 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 Destination	Mirror Direction	Unblocked Vlan	Config Status	Oper Status
6.	2/10	-	NONE	Enable	On
-----+-----+-----+-----+-----+-----					
	Mirror Source				
-----+-----+-----+-----+-----+-----					
6.	2/3	bidirectional	-	Enable	On
6.	2/4	bidirectional	-	Enable	On
6.	2/5	bidirectional	-	Enable	On
6.	2/6	bidirectional	-	Enable	On
6.	2/7	bidirectional	-	Enable	On
6.	2/8	bidirectional	-	Enable	On
6.	2/9	bidirectional	-	Enable	On

For more information about this command, see [“Displaying Port Mirroring Status” on page 26-21](#) or the [“Port Mirroring and Monitoring Commands” chapter in the *OmniSwitch 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 26-23.

Port Monitoring Specifications

Platforms Supported	OS10K
Ports Supported	Ethernet (10 Mbps) Fast Ethernet (100 Mbps) Gigabit Ethernet (1000 Mbps) 10 Gigabit Ethernet (10000 Mbps)
Monitoring Sessions Supported	One per switch
File Type Supported	ENC file format (Network General Sniffer Network Analyzer Format)

Port Monitoring Defaults

The following table shows port mirroring default values.

Global Port Monitoring Defaults

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

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 slot and port number of the port to be monitored. For example:

```
-> port-monitoring 6 source 2/3
```

- 2 Enable the port monitoring session by entering **port-monitoring**, followed by the port monitoring session ID, **source**, the slot and port number of the port to be monitored, and **enable**. For example:

```
-> port-monitoring 6 source 2/3 enable
```

- 3 *Optional.* Configure optional parameters. For example, to create a file called “monitor1” for port monitoring session 6 on port 2/3, enter:

```
-> port-monitoring 6 source 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. Src	Mon Dir	Over write	Oper. Stat	Admin Stat	Capt. Type	Max. Size	File Name
6.	2/ 3	Bidirectional	ON	ON	ON	brief		

For more information about this command, see [“Port Monitoring” on page 26-23](#) or the “Port Mirroring and Monitoring Commands” chapter in the *OmniSwitch 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 26-29.

sFlow Specifications

RFCs Supported	3176 - sFlow Management Information Base
Platforms Supported	OS10K
Sampling	Sampling rate of one (1) counts all packets and 0 (zero) disables sampling.
Agent IP Address	Configurable using sflow agent ip command.

sFlow Defaults

The following table shows sFlow default values:

sFlow Defaults

Parameter Description	CLI Command	Default Value/Comments
Receiver Name	sflow agent	Empty
Timeout Value	sflow agent	0 seconds
IP Address	sflow agent	32 bit address (IPv4)
Data File Size	sflow agent	1400 Bytes
Version Number	sflow agent	5
Destination Port	sflow agent	6343
Receiver Index	sflow sampler	0
Packet Sampling Rate	sflow sampler	0
Sampled Packet Size	sflow sampler	128 Bytes
Receiver Index	sflow poller	0
Interval Value	sflow poller	0 seconds

Quick Steps for Configuring sFlow

Follow the steps below to create a sFlow receiver session.

- 1 To create a sFlow receiver session, use the **sflow agent** command by entering **sflow receiver**, followed by the receiver index, name, and the address to be monitored. 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 26-29 or the “sFlow Commands” chapter in the *OmniSwitch 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 2/1-5 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 2/1-5, enter:

```
-> sflow sampler 1 2/1-5 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        2048  128
1          2/ 2           1        2048  128
1          2/ 3           1        2048  128
1          2/ 4           1        2048  128
1          2/ 5           1        2048  128
```

For more information about this command, see “sFlow” on page 26-29 or the “sFlow Commands” chapter in the *OmniSwitch 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 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
1	2/ 6	1	30
1	2/ 7	1	30
1	2/ 8	1	30
1	2/ 9	1	30
1	2/10	1	30

For more information about this command, see “sFlow” on page 26-29 or the “sFlow Commands” chapter in the *OmniSwitch 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 26-34](#).

RMON Specifications

RFCs Supported	2819 - Remote Network Monitoring Management Information Base
Platforms Supported	OS10K
RMON Functionality Supported	Basic RMON 4 group implementation –Ethernet Statistics group –History (Control and Statistics) group –Alarms group –Events group
RMON Functionality Not Supported	RMON 10 group* RMON2* –Host group –HostTopN group –Matrix group –Filter group –Packet Capture group (*An external RMON probe that includes RMON 10 group and RMON2 be used where full RMON probe functionality is required.)
Flavor (Probe Type)	Ethernet/History/Alarm
Status	Active/Creating/Inactive
History Control Interval (seconds)	1 to 3600
History Sample Index Range	1 to 65535
Alarm Interval (seconds)	1 to 2147483647
Alarm Startup Alarm	Rising Alarm/Falling Alarm/ RisingOrFalling Alarm
Alarm Sample Type	Delta Value/Absolute
RMON Traps Supported	RisingAlarm/FallingAlarm These traps are generated whenever an Alarm entry crosses either its Rising Threshold or its Falling Threshold and generates an event configured for sending SNMP traps.

RMON Probe Defaults

The following table shows Remote Network Monitoring default values.

Global RMON Probe Defaults

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
```

The display is similar to the one shown below:

```
Entry  Slot/Port  Flavor  Status  Duration  System Resources
-----+-----+-----+-----+-----+-----
1011   1/11   Ethernet Active   11930:27:05  272 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 1011
```

The display appears similar to the one shown below:

```
Probe's Owner: Switch Auto Probe on Slot 1, Port 11
Entry 1011
  Flavor = Ethernet, Status = Active,
  Time = 11930 hrs 26 mins,
  System Resources (bytes) = 272
```

For more information about these commands, see [“Displaying a List of RMON Probes”](#) on page 26-37, [“Displaying Statistics for a Particular RMON Probe”](#) on page 26-38, or the [“RMON Commands”](#) chapter in the *OmniSwitch 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 26-41](#).

Switch Health Specifications

Platforms Supported	OS10K
Health Functionality Supported	<ul style="list-style-type: none">–Switch level CPU Utilization Statistics (percentage);–Switch/module/port level Input Utilization Statistics (percentage);–Switch/module/port level Input/Output Utilization Statistics (percentage);–Switch level Memory Utilization Statistics (percentage);–Device level (e.g., Chassis/CMM) Temperature Statistics (Celsius).
Monitored Resource Utilization Levels	<ul style="list-style-type: none">–Most recent utilization level;–Average utilization level during last minute;–Average utilization level during last hour;–Maximum utilization level during last hour.
Resource Utilization Raw Sample Values	Saved for previous 60 seconds.
Resource Utilization Current Sample Values	Stored.
Resource Utilization Maximum Utilization Value	Calculated for previous 60 seconds and stored.
Utilization Value = 0	Indicates that none of the resources were measured for the period.
Utilization Value = 1	Indicates that a non-zero amount of the resource (less than 2%) was measured for the period.
Percentage Utilization Values	Calculated based on Resource Measured During Period/Total Capacity.
Resource Threshold Levels	Apply automatically across all levels of switch (switch/module/port).
Rising Threshold Crossing	A Resource Threshold was exceeded by its corresponding utilization value in the current cycle.
Falling Threshold Crossing	A Resource Threshold was exceeded by its corresponding utilization value in the previous cycle, but is not exceeded in the current cycle.
Threshold Crossing Traps Supported	Device, module, port-level threshold crossings.

Switch Health Defaults

The following table shows Switch Health default values.

Global Switch Health Defaults

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 26-44 or the [“Health Monitoring Commands”](#) chapter in the *OmniSwitch 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.

Note the following restriction when configuring a port mirroring session:

- Two (2) port mirroring sessions are supported per standalone chassis-based switch.
- You can configure maximum two sessions (any combination) on the same NI module regardless of type (C48,U48, or X32).

What Ports Can Be Mirrored?

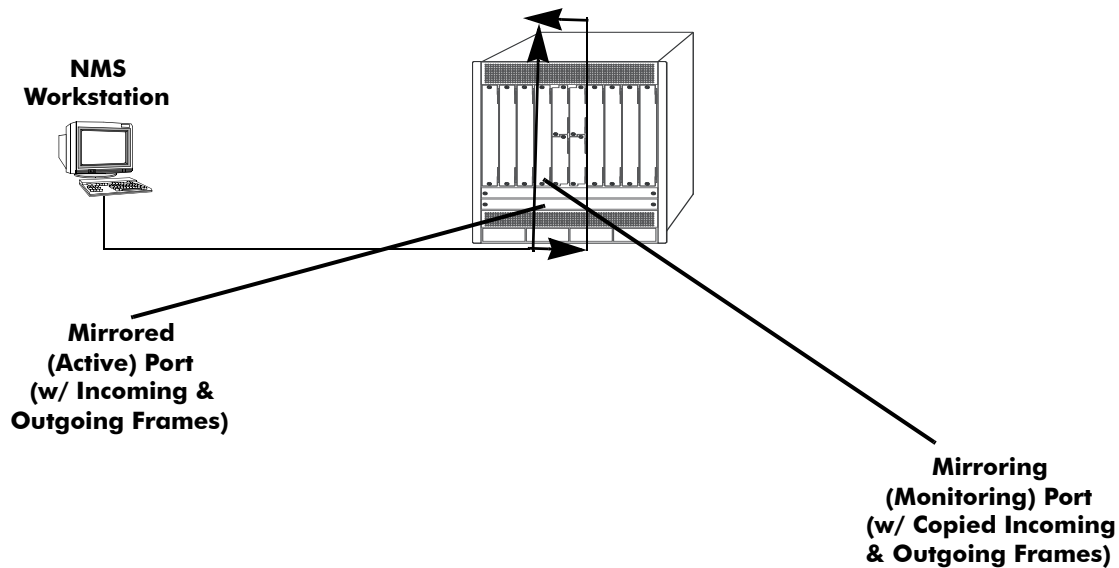
Mirroring between any similar ports and between any SFP to any other SFP port is supported.

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 be some performance degradation, since all frames received and transmitted by the mirrored port need to be copied and sent to the mirroring port.



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

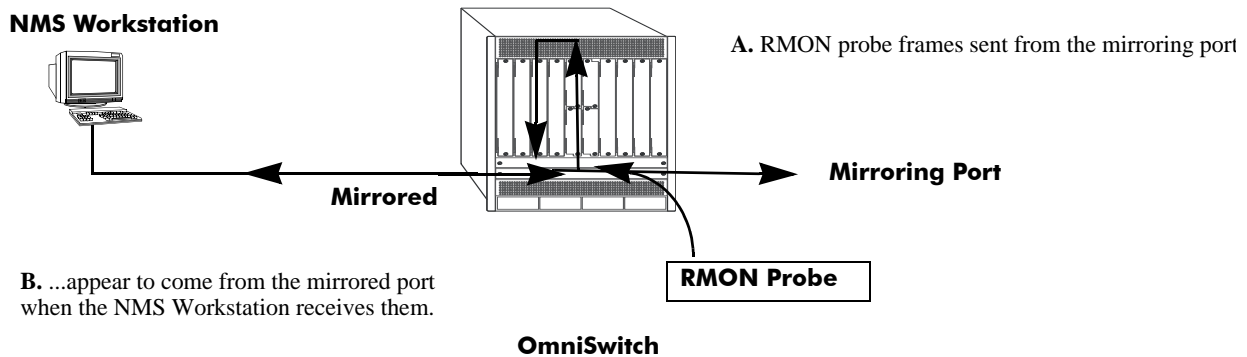
If mirroring is enabled on multiple ports and the same traffic is passing through these ports, then only one copy of each packet is sent to the mirroring destination. When the packet is mirrored for the first time, the switching ASIC flags the packet as “already mirrored” If the packet goes through one 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

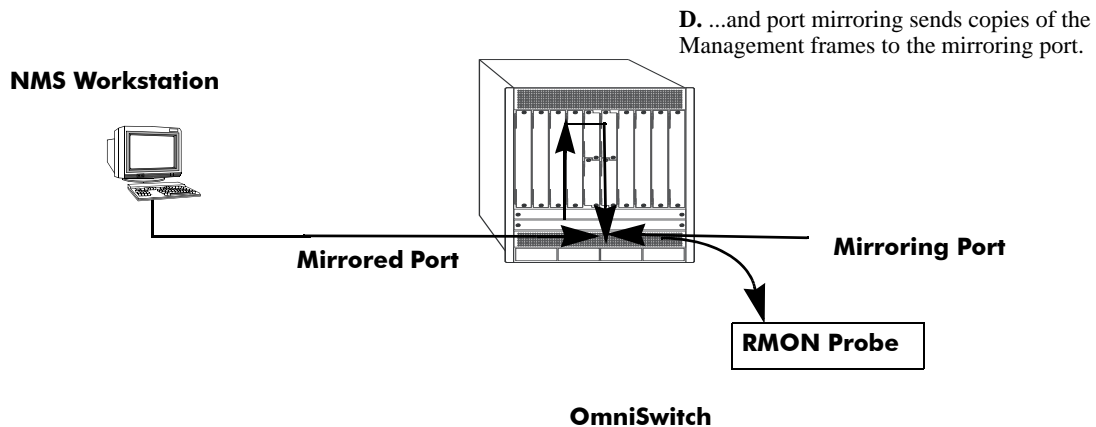
Port mirroring is a helpful monitoring tool when used in conjunction with an external RMON probe. Once you set up port mirroring, the probe can collect all relevant RMON statistics for traffic on the mirrored port. You can also move the mirrored port so that the mirroring port receives data from different 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 26-19 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.



C. Management frames from the NMS Workstation are sent to the mirrored port...



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)
- BPDUs are not mirrored on OmniSwitch 10K switches-

For more information and an example of a Remote Port Mirroring configuration, see [“Remote Port Mirroring” on page 26-17](#).

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 slot/ports, as shown in the following example:

```
-> port-mirroring 6 source 2/3 destination 2/4
```

This command line specifies mirroring session 6, with the source (mirrored) port located in slot 2/port 3, and the destination (mirroring) port located in 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 slot/ports, and the remote port mirroring VLAN ID as shown in the following example:

```
-> port-mirroring 8 source 1/1 destination 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 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 2/4 in session 1:

```
-> port-mirroring 1 source 1/2 destination 2/4
-> port-mirroring 1 source 2/1 destination 2/4
-> port-mirroring 1 source 2/3 destination 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 2/4 in session 1:

```
-> port-mirroring 1 source 1/2-6 destination 2/4
```

In the following example, ports 1/9, 2/7, and 3/5 are mirrored on destination port 2/4 in session 1:

```
-> port-mirroring 1 source 1/9 2/7 3/5 destination 2/4
```

In the following example, 1/2 through 1/6 and 1/9, 2/7, and 3/5 are mirrored on destination port 2/4 in session 1:

```
-> port-mirroring 1 source 1/2-6 1/9 2/7 3/5 destination 2/4
```

Note. Ports can be added after a port mirroring session has been configured.

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 2/3 destination 2/4 unblocked-vlan 750
```

This command line specifies mirroring session 6, with the source (mirrored) port located in slot 2/port 3, and the destination (mirroring) port located in 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 26-20 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 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 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 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 2/3 destination 6/4 inport
```

In this example, the command specifies port mirroring session 6, with the mirrored (active) port located in slot 2/port 3, and the mirroring port located in slot 6/port 4. The mirroring direction is unidirectional and outward bound:

```
-> port-mirroring 6 source 2/3 destination 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 2/3 destination 6/4 bidirectional
```

Note. The port mirroring session identifier and slot/port locations of the designated interfaces must always be specified.

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 Destination	Mirror Direction	Unblocked Vlan	Config Status	Oper Status
1.	2/1	-	NONE	Enable	On
Mirror Source					
1.	1/1	bidirectional	-	Enable	On
1.	1/2	bidirectional	-	Enable	On
1.	1/3	bidirectional	-	Enable	On
1.	1/4	bidirectional	-	Enable	On
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
```

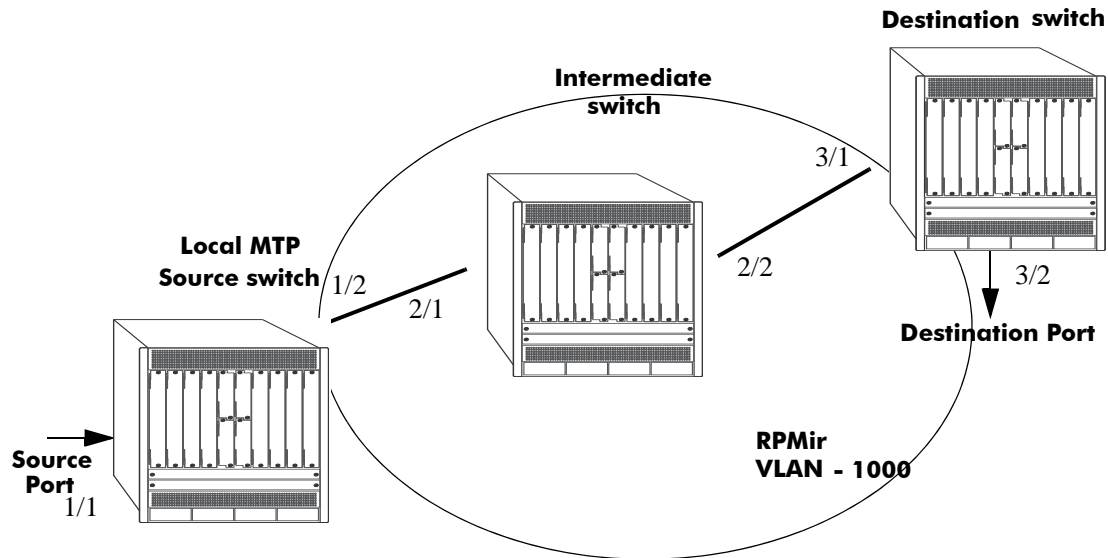
In this example, port mirroring session 6 is deleted.

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:



Remote Port Mirroring Example

Configuring Source Switch

Follow the steps given below to configure the Source Switch:

```
-> port-mirroring 8 source 1/1
-> port-mirroring 8 destination 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 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 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 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 rule r_ds1 condition c_ds1 action a_ds1
-> qos apply
```

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.
- You cannot configure a port mirroring and a port monitoring session on the same NI module in an OmniSwitch chassis-based switch.
- You cannot configure port mirroring and monitoring on the same switching ASIC on OmniSwitch 10K switches. Each switching ASIC controls 24 ports (e.g., ports 1–24, 25–48, etc.). For example, if a port mirroring session is configured for ports 1/12 and 1/22, then configuring a port monitoring session for any of the ports between 1 and 24 is not allowed.
- If a port mirroring session is configured across two switching ASICs, then configuring a monitoring session is not allowed on any of the ports controlled by each of the ASICs involved. For example, if a

port mirroring session is configured for ports 1/8 and 1/30 on a 48-port switch, then configuring a port monitoring session involving any of the ports between 1 and 48 is not allowed.

- 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 26-15 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 2/3 enter:

```
-> port-monitoring 6 source 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 2/3 and administratively enable it, enter:

```
-> port-monitoring 6 source 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 2/3, enter:

```
-> port-monitoring 6 source 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 2/3 that lasts 12000 seconds before it is disabled, enter:

```
-> port-monitoring 6 source 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 2/3 with a data file called “user_port” in the **/flash** directory, enter:

```
-> port-monitoring 6 source 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 does 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 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 2/3 file /flash/user_port size 3
```

To select the 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 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 2/3 file /flash/user_port 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 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 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 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 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 2/3 as unidirectional and inward bound, enter:

```
-> port-monitoring 6 source 2/3 inport
```

To configure port monitoring session 6 on port 2/3 as unidirectional and outward bound, for example, enter:

```
-> port-monitoring 6 source 2/3 outport
```

For example, to restore port monitoring session 6 on port 2/3 to its bidirectional direction, enter:

```
-> port-monitoring 6 source 2/3 bidirectional
```

Configuring capture-type

To configure the amount of data to be captured, use the **port-monitoring source capture-type** command. If the mode of capture-type is set to 'brief', only the first 64 bytes of packets will be captured. If the mode of capture-type 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 2/3 to capture only the first 64 bytes of the packet, enter

```
-> port-monitoring 6 source 2/3 capture-type brief
```

To configure port monitoring session 6 on port 2/3 to capture full packet, enter

```
-> port-monitoring 6 source 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 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 a 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 from the Q-Dispatcher module. The counter samples are given to the poller module and sampled packets are given to the sampler to format a UDP. The sFlow manager also has a timer which periodically sends timer ticks to other sections.

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 destination 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 gets hardware sampled from Q-Dispatcher 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 agent** command by entering **sflow receiver**, followed by the receiver_index, name, the name of the session and **address**, and the IP address of the switch to be monitored.

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 time-out 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 2/3, enter:

```
-> sflow sampler 1 port 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 2/3 and to specify the rate and sample-hdr-size, enter:

```
-> sflow sampler 1 port 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, enter:

```
-> sflow poller 3 port 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 with an interval of 5, enter:

```
-> sflow poller 3 port 1/1 receiver 6 interval 5
```

Configuring a Fixed Primary Address

In order to generate the IP packets and send the sFlow data-grams out into the network, sFlow agent requires an IP address configured to it. The agent's IP address can be configured using the **sflow agent ip** command. If there are 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 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	Sample-rate	Sample-hdr-size
1	2/ 1	1	2048	128
1	2/ 2	1	2048	128
1	2/ 3	1	2048	128
1	2/ 4	1	2048	128
1	2/ 5	1	2048	128

Note. For more information about the displays that result from these commands, see the *OmniSwitch 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
1	2/ 6	1	30
1	2/ 7	1	30
1	2/ 8	1	30
1	2/ 9	1	30
1	2/10	1	30

Note. For more information about the displays that result from these commands, see the *OmniSwitch 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:

```
-> ip interface Loopback0 127.0.0.1
-> 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 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 2/3
```

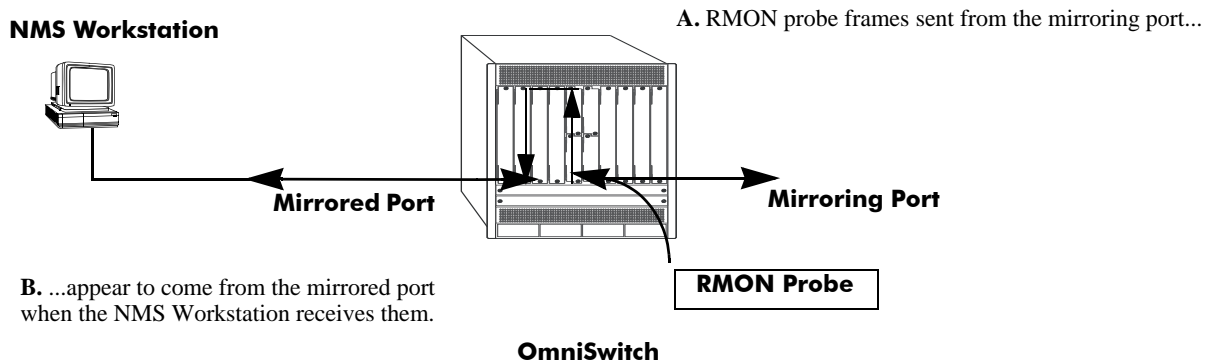
To delete a sFlow poller session, use the no form of the **sflow poller** command by entering **no sflow poller**, followed by the instance ID number, the slot number of the port to delete, a slash (/), and the port number of the port, enter:

```
-> no sflow poller 3 port 1/1
```

Remote Monitoring (RMON)

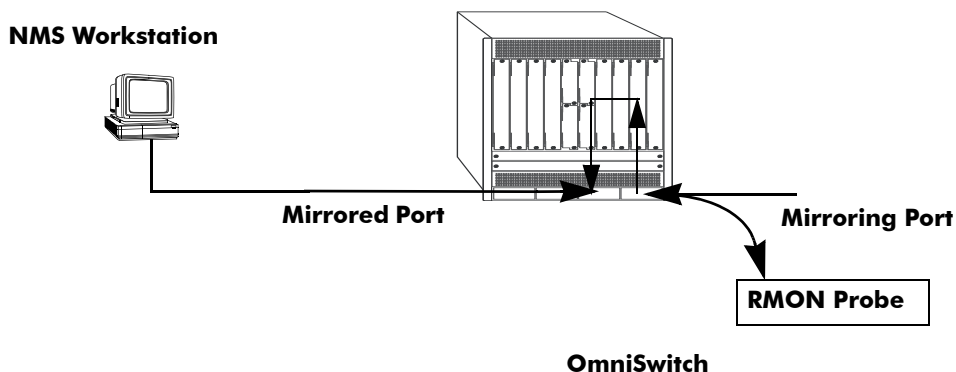
Remote Network Monitoring (RMON) is an SNMP protocol used to manage networks remotely. *RMON probes* can be used to collect, interpret, and forward statistical data about network traffic from designated active ports in a LAN segment to an NMS (Network Management System) application for monitoring and 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.



B. ...appear to come from the mirrored port when the NMS Workstation receives them.

C. Management frames from the NMS Workstation are sent to the mirrored port...



D. ...and port mirroring sends copies of the Management frames to the mirroring port.

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
```

A display showing all current statistics RMON probes must appear, as shown in the following example:

Entry	Slot/Port	Flavor	Status	Duration	System Resources
4001	4/1	Ethernet	Active	00:25:00	275 bytes
4008	4/8	Ethernet	Active	00:25:00	275 bytes
4005	4/5	Ethernet	Active	00:25:00	275 bytes

This table entry displays probe statistics for all probes on the switch. The probes are active, utilize 275 bytes of memory, and 25 minutes have elapsed since the last change in status occurred.

To show a list of the history probes, enter:

```
-> show rmon probes history
```

A display showing all current history RMON probes must appear, as shown in the following example:

Entry	Slot/Port	Flavor	Status	Duration	System Resources
1	1/1	History	Active	92:52:20	5464 bytes
30562	1/35	History	Active	00:31:22	312236 bytes
30817	1/47	History	Active	00:07:31	5200236 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
```

A display showing all current alarm RMON probes must appear, as shown in the following example:

Entry	Slot/Port	Flavor	Status	Duration	System Resources
31927	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 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 (275).

```
-> show rmon probes 4005
```

```
Probe's Owner: Switch Auto Probe on Slot 4, Port 5
Entry 4005
Flavor = Ethernet, Status = Active
Time = 48 hrs 54 mins,

System Resources (bytes) = 275
```

Sample Display for History Probe

The display shown here identifies RMON Probe 10325's Owner description and interface location (Analyzer-p:128.251.18.166 on slot 1, port 35), the total number of History Control Buckets (samples) requested and granted (2), along with the time interval for each sample (30 seconds) and system-generated Sample Index ID number (5859). The probe Entry number identifier (10325), probe Flavor (History), and Status (Active), the amount of time that has elapsed since the last change in status (48 hours, 53 minutes), and the amount of memory allocated to the probe, measured in bytes (601) are also displayed.

```
-> show rmon probes history 30562

Probe's Owner: Analyzer-p:128.251.18.166 on Slot 1, Port 35

History Control Buckets Requested    = 2
History Control Buckets Granted      = 2
History Control Interval              = 30 seconds
History Sample Index                  = 5859
Entry 10325
    Flavor = History, Status = Active
    Time = 48 hrs 53 mins,
    System Resources (bytes) = 601
```

Sample Display for Alarm Probe

The display shown here identifies RMON Probe 11235's Owner description and interface location (Analyzer-t:128.251.18.166 on slot 1, port 35), 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: Analyzer-t:128.251.18.166 on Slot 1, Port 35
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
```

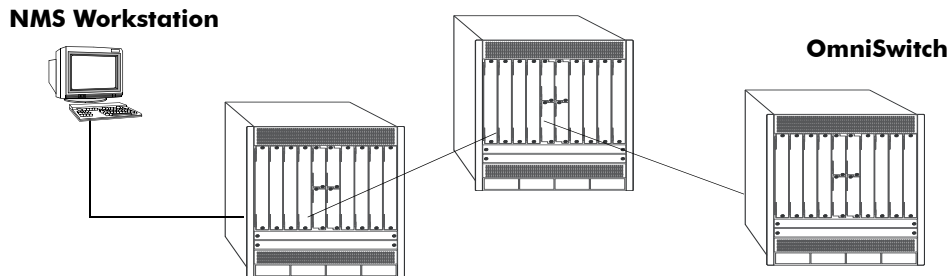
A display showing the specific logged RMON Event must appear, as shown in the following example:

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.



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 26-43](#) for more information.
- **show health configuration**—Displays current health threshold settings. See [page 26-44](#) for details.
- **health interval**—Configures sampling interval between health statistics checks. See [page 26-45](#) for more information..
- **show health** —Displays health statistics for the switch, as percentages of total resource capacity. See [page 26-46](#) 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 <i>incoming traffic only</i> . The total bandwidth is the Ethernet port capacity of <i>all NI modules</i> 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 <i>all incoming and outgoing traffic</i> . 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 36, “Health Monitoring Commands,”](#) in the *OmniSwitch 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 26-43, as well as [Chapter 36, “Health Monitoring Commands,”](#) in the *OmniSwitch 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    5    4    8
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 4/3
* - current value exceeds threshold

Port 04/03
Resources          Limit      Curr      1 Min      1 Hr      1 Hr
                  +-----+ +-----+ +-----+ +-----+ +-----+
                  |         | |         | |         | |         | |         |
Receive            80      01      01      01      01
Transmit/Receive  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.

27 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 CLI Reference Guide*.

This chapter provides an overview of VLAN Stacking and includes the following topics:

- [“VLAN Stacking Specifications” on page 27-2.](#)
- [“VLAN Stacking Defaults” on page 27-2.](#)
- [“VLAN Stacking Overview” on page 27-3.](#)
- [“Interaction With Other Features” on page 27-7.](#)
- [“Configuring VLAN Stacking Services” on page 27-10](#)
- [“VLAN Stacking Application Example” on page 27-20..](#)
- [“Verifying the VLAN Stacking Configuration” on page 27-24](#)
- [“Verifying the VLAN Stacking Configuration” on page 27-24.](#)

VLAN Stacking Specifications

Platforms Supported	OmniSwitch 10K
IEEE Standards Supported	IEEE 802.1Q, 2003 Edition, IEEE Standards for Local and metropolitan area networks—Virtual Bridged Local Area Networks P802.1ad/D6.0 (C/LM) Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks - Amendment 4: Provider Bridges
Maximum number of Services	4K
Maximum number of SVLANs	4K
Maximum number of UNI ports	N/A
Maximum number of NNI ports	N/A
Maximum number of SAPs	1K
Maximum number of SAP Profiles	1K

VLAN Stacking Defaults

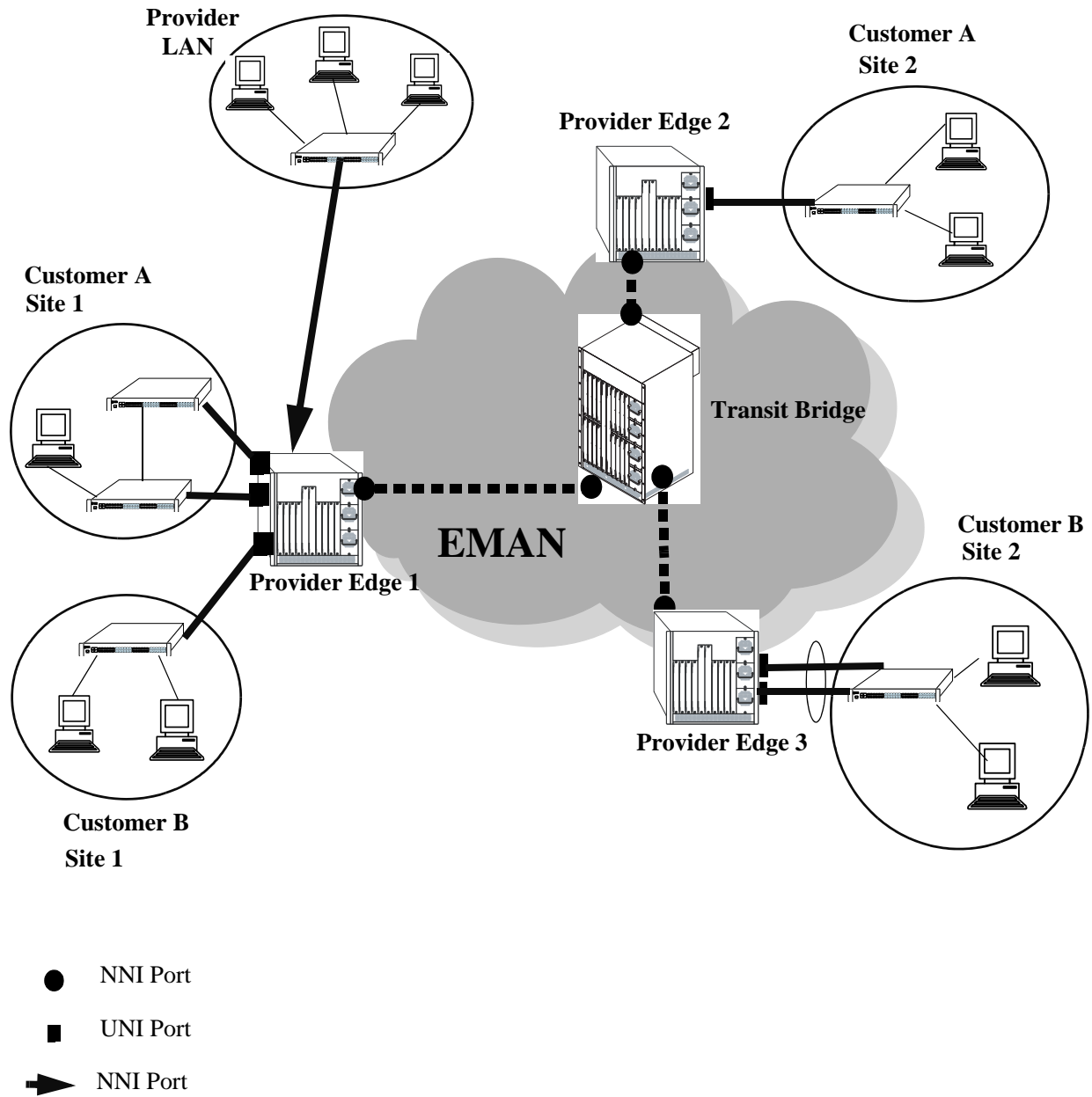
Parameter Description	Command	Default Value/Comments
SVLAN administrative and Spanning Tree status.	ethernet-service svlan	Enabled
Vendor TPID and legacy BPDU support for STP on a VLAN Stacking network port.	ethernet-service nni	TPID = 0x8100 legacy STP BPDU = dropped.
Acceptable frame types on a VLAN 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: 802.3ad, UDLD, OAM, LACP-Marker Tunneled Frames: STP, MVRP, Discarded Frames: 802.1ab, 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:



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	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 27-10](#) 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:

- QoS allocates switch resources for VLAN Stacking Service attributes, even though such attributes are not configurable via the QoS CLI.
- 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.
- Quarantine Manager and Remediation (QMR) is not available if VLAN Stacking services or QoS **inner source vlan** and **inner 802.1p** policies are configured on the switch.

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 27-19](#) for information about configuring VLAN Stacking to tunnel or discard Spanning Tree BPDU.
- See [“Configuring VLAN Stacking Network Ports” on page 27-13](#) 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 3/1 as a VLAN Stacking Network Network Interface (NNI) port and associate the port with SVLAN 1001 using the **ethernet-service svlan nni** command.

```
-> ethernet-service svlan 10 nni port 1/3
```

- 4 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
```

- 5 Configure port 1/49 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 1/49
```

- 6 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
```

- 7 (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
```

- 8 (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
```

- 9 (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
```

- 10 (Optional) Associate the “uni_1” profile with port 1/49 using the **ethernet-service uni uni-profile** command.

```
-> ethernet-service uni port 1/49 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)    : 2/1, 3/2
  SAP Id    : 20
    UNIs     : 1/1, 1/2
    CVLAN(s) : 10, 20
    sap-profile : sap-video1
  SAP Id    : 30
    UNIs     : 1/3
    CVLAN(s) : untagged, 40
    sap-profile : sap-video2
```

```
Service Name : CustomerABC
  SVLAN      : 255
  NNI(s)    : 1/22
  SAP Id    : 10
    UNIs     : 2/10, 2/11
    CVLAN(s) : 500, 600
    sap-profile : default-sap-profile
```

```
-> show ethernet-service service-name CustomerABC
```

```
Service Name : CustomerABC
  SVLAN      : 255
  NNI(s)    : 1/22
  SAP Id    : 10
    UNIs     : 2/10, 2/11
    CVLAN(s) : 500, 600
    sap-profile : default-sap-profile
```

```
-> show ethernet-service svlan 300
```

```
Service Name : VideoOne
  SVLAN      : 300
  NNI(s)    : 2/1, 3/2
  SAP Id    : 20
    UNIs     : 1/1, 1/2
    CVLAN(s) : 10, 20
    sap-profile : sap-video1
  SAP Id    : 30
    UNIs     : 1/3
    CVLAN(s) : 30, 40
    sap-profile : sap-video2
```

See the *OmniSwitch 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 27-8](#).

- 1 Create an SVLAN.** An SVLAN is associated to a VLAN Stacking service to carry customer or provider traffic. See [“Configuring SVLANs” on page 27-11](#).
- 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 27-12](#).
- 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 27-13](#).
- 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 27-14](#).
- 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 27-15](#).
- 6 Associate CVLAN traffic with an 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 27-16](#).
- 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 27-17](#).
- 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 27-19](#).

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 27-11 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)     : 2/1, 3/2
  SAP Id     : 20
    UNIs      : 1/1, 1/2
    CVLAN(s)  : 10, 20
    sap-profile : sap-video1
  SAP Id     : 30
    UNIs      : 1/3
    CVLAN(s)  : untagged, 40
    sap-profile : sap-video2

Service Name : CustomerABC
  SVLAN      : 255
  NNI(s)     : 1/22
  SAP Id     : 10
    UNIs      : 2/10, 2/11
    CVLAN(s)  : 500, 600
    sap-profile : default-sap-profile
```

```
-> show ethernet-service service-name CustomerABC
```

```
Service Name : CustomerABC
  SVLAN      : 255
  NNI(s)     : 1/22
  SAP Id     : 10
    UNIs      : 2/10, 2/11
    CVLAN(s)  : 500, 600
    sap-profile : default-sap-profile
```

```
-> show ethernet-service svlan 300
```

```
Service Name : VideoOne
SVLAN       : 300
NNI(s)      : 2/1, 3/2
SAP Id      : 20
  UNIs       : 1/1, 1/2
  CVLAN(s)   : 10, 20
  sap-profile : sap-video1
SAP Id      : 30
  UNIs       : 1/3
  CVLAN(s)   : 30, 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 svlan nni** command is used to configure a switch port or link aggregate of ports as a VLAN Stacking Network Network Interface (NNI) and associate the NNI with an SVLAN. For example, the following command configures port 2/1 as an NNI port and associates 2/1 with SVLAN 300:

```
-> ethernet-service svlan 300 nni port 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. In addition, 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.

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 2/1
```

Note that when the last SVLAN association for the port is deleted, the port automatically reverts back to a conventional switch port and is no longer VLAN Stacking capable.

Use the **show ethernet-service** command to verify the NNI port configuration for the switch.

Configuring NNI Port Parameters

The **ethernet-service nni** command is used to configure the following parameters that apply to traffic processed by NNI ports:

- **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 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 with normal VLANs. The TPID of the packets tagged with the normal VLAN is 0x8100 (regardless of the TPID of the NNI port). This allows the NNI port to carry both 802.1q tagged traffic and SVLAN tagged traffic.
- Only fixed ports can be converted to NNI ports.

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). An 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 27-15.](#)
- **Customer VLANs (CVLANs).** See [“Configuring the Type of Customer Traffic to Tunnel” on page 27-16.](#)
- **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 27-17.](#)

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 27-17](#). 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 as an UNI port and associates 1/1 with SAP 20:

```
-> ethernet-service sap 20 uni port 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 1/1 and SAP 20:

```
-> no ethernet-service sap 20 uni port 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 27-16](#).
- 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. While Spanning Tree frames are automatically tunneled, all other protocol control frames are dropped.
- To use different profile attribute values, create a new profile and associate it with the UNI port. See [“Configuring a UNI Profile” on page 27-19](#). 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 is 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.

Profile Attribute	Command Parameters	Description
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 27-14](#)). 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-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
```

Note the following when associating a profile with a VLAN Stacking SAP:

- 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

The **ethernet-service uni-profile** command is used to create a VLAN Stacking UNI port profile. The UNI profile determines how control frames ingressing on UNI ports are processed. For example, the following command creates a UNI profile to specify that VLAN Stacking should discard MVRP frames:

```
-> ethernet-service uni-profile uni_1 l2-protocol mvrp discard
```

A default UNI profile, named “default-uni-profile”, is automatically associated with a UNI port. The default UNI profile specifies how control frames ingressing on the UNI port.

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 view a list of profiles that are already configured for the switch. This command also displays the attribute values for each 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:

```
-> ethernet-service uni port 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 uni-profile default-uni-profile
```

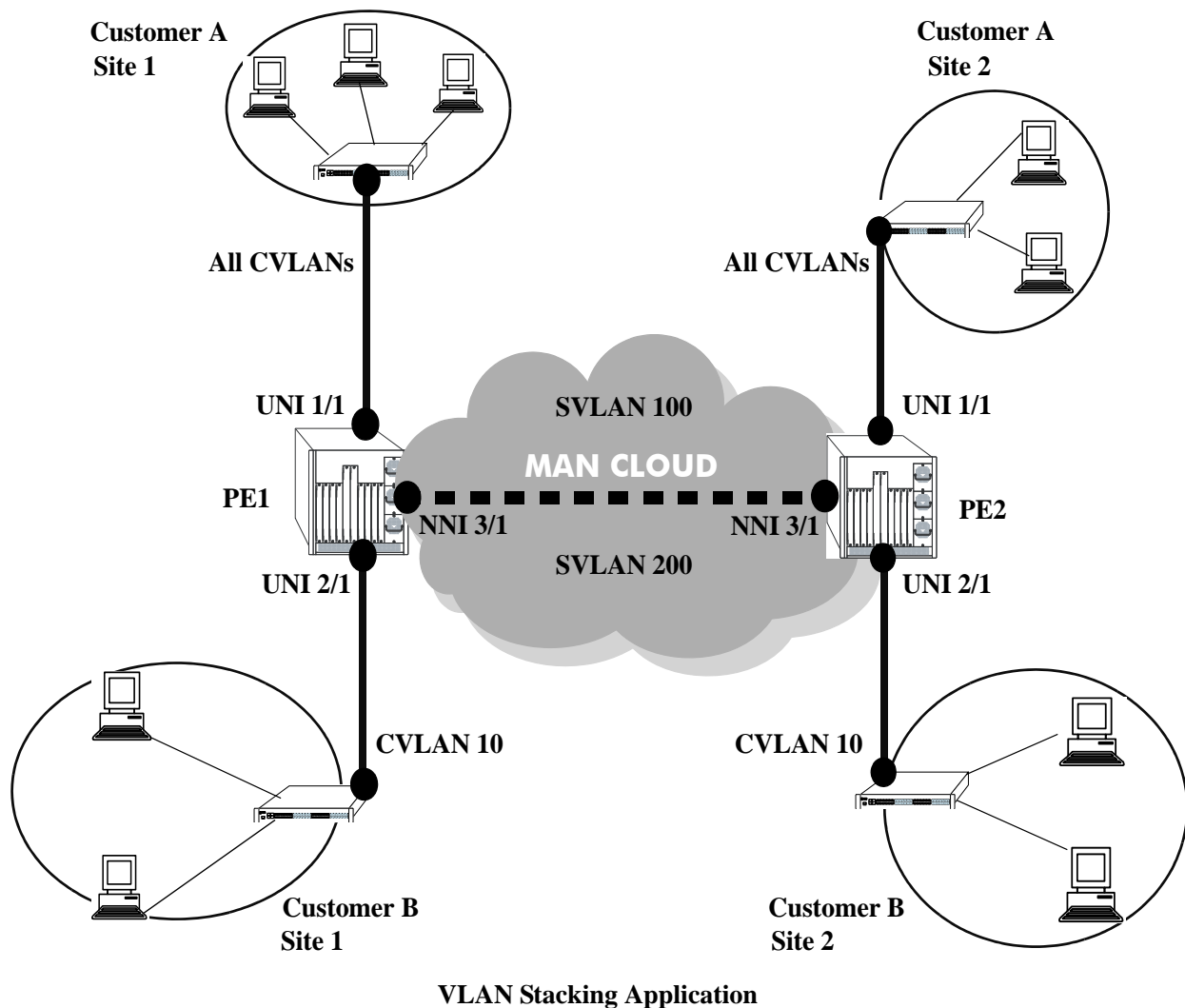
Use the **show ethernet-service uni** command to display the profile associations for each UNI port.

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 through 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.



VLAN Stacking Configuration Example

This section provides a tutorial for configuring the sample application, as illustrated on [page 27-20](#), 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 3/1 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 3/1
-> ethernet-service svlan 200 nni port 3/1
```

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 on PE1 *and* PE2 as a VLAN Stacking UNI port and associate 1/1 with SAP 20 using the **ethernet-service sap uni** command.

```
-> ethernet-service sap 20 uni port 1/1
```

7 Configure port 2/1 on PE1 *and* PE2 as a VLAN Stacking UNI port and associate 2/1 with SAP 30 using the **ethernet-service sap uni** command.

```
-> ethernet-service sap 30 uni port 2/1
```

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)     : 3/1
  SAP Id     : 20
    UNIs      : 1/1
    CVLAN(s)  : all
  sap-profile : default-sap-profile
```

```
Service Name : CustomerB
  SVLAN      : 200
  NNI(s)     : 3/1
  SAP Id     : 10
    UNIs      : 2/1
    CVLAN(s)  : 10
  sap-profile : map_pbit
```

```
-> show ethernet-service
```

```
Service Name : CustomerABC
  SVLAN      : 255
  NNI(s)     : 1/22
  SAP Id     : 10
    UNIs      : 2/10, 2/11
    CVLAN(s)  : 500, 600
  sap-profile : default-sap-profile
```

```
-> show ethernet-service service-name CustomerABC
```

```
Service Name : CustomerABC
  SVLAN      : 255
  NNI(s)     : 1/22
  SAP Id     : 10
    UNIs      : 2/10, 2/11
    CVLAN(s)  : 500, 600
  sap-profile : default-sap-profile
```

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 3/1
```

```
-> ethernet-service sap 20 service-name CustomerA
-> ethernet-service sap 20 uni 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 3/1
-> ethernet-service sap 30 service-name CustomerB
-> ethernet-service sap 30 uni 2/1
-> 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
```

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:

show ethernet-service vlan	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.

For more information about the resulting displays from these commands, see the *OmniSwitch 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 27-8.

28 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 CLI Reference Guide*.

In This Chapter

The following procedures are described:

- [“Enabling Switch Logging” on page 28-6](#)
- [“Setting the Switch Logging Severity Level” on page 28-6](#)
- [“Specifying the Switch Logging Output Device” on page 28-9](#)
- [“Displaying Switch Logging Records” on page 28-12](#)

Notes. 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 Customer Service representative for assistance with debugging functions.

Switch Logging Specifications

Platforms Supported	OmniSwitch 10K
Functionality Supported	High-level event logging mechanism that forwards requests from applications to enabled logging devices.
Functionality Not Supported	Not intended for debugging individual hardware applications.
Logging Devices	Flash Memory/Console/IP Address
Application ID Levels Supported	<p>IDLE (255), DIAG (0), IPC-DIAG (1), QDRIVER (2), QDISPATCHER (3), IPC-LINK (4), NI-SUPERVISION (5), INTERFACE (6), 802.1Q (7), VLAN (8), GM (9), BRIDGE (10), STP (11), LINKAGG (12), QOS (13), RSVP (14), IP (15), IPMS (17), AMAP (18), GMAP (19), SLB(25), AAA (20), IPC-MON (21), IP-HELPER (22), PMM (23), MODULE (24), EIPC (26), CHASSIS (64), PORT-MGR (65), CONFIG (66), CLI (67), SNMP (68), WEB (69), MIPGW (70), SESSION (71), TRAP (72), POLICY (73), DRC (74), SYSTEM (75), HEALTH (76), NAN-DRIVER (78), RMON (79), TELENET (80), PSM (81), FTP (82), SNMI (83), DISTRIB (84), EPILOGUE (85), LDAP (86), NOSNMP (87), SSL (88), DBGGW (89), LANPOWER (108)</p>
Severity Levels/Types Supported	<p>2 (Alarm - highest severity), 3 (Error), 4 (Alert), 5 (Warning) 6 (Info - default), 7 (Debug 1), 8 (Debug 2), 9 (Debug 3 - lowest severity)</p>

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 level	Default severity level is info. The numeric equivalent for info is 6
Enabling/Disabling switch logging Output	swlog output	Flash Memory and Console
Switch logging file size	swlog output flash file-size	TBD

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           : 100K bytes,
Log Device                   : console flash,
Syslog FacilityID           : local0(16),
Remote command-log          : Disabled,
Hash Table entries age limit : 60 seconds,
Switch Log Preamble         : Enabled,
Switch Log Debug             : Disabled,
Switch Log Duplicate Detection : Enabled,
Console Display Level       : debug3,
```

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.

Notes. 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, while **no swlog** disables it.

To enable switch logging, enter the **swlog** command:

```
-> swlog
```

To disable switch logging, enter the **no swlog** command:

```
-> no swlog
```

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 level** command is used to assign the severity levels to the applications.

The syntax for the **swlog appid level** 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.

The application ID information is shown in the following table. The severity level information is shown in the table beginning on [page 28-8](#).

CLI Keyword	Numeric Equivalent	Application ID
IDLE	255	APPID_IDLE
DIAG	0	APPID_DIAGNOSTICS
IPC-DIAG	1	APPID_IPC_DIAGNOSTICS
QDRIVER	2	APPID_QDRIVER
QDISPATCHER	3	APPID_QDISPATCHER
IPC-LINK	4	APPID_IPC_LINK
NI-SUPERVISION	5	APPID_NI_SUP_AND_PROBER
INTERFACE	6	APPID_ESM_DRIVER
802.1Q	7	APPID_802.1Q
VLAN	8	APPID_VLAN_MGR
GM	9	APPID_GROUPMOBILITY (RESERVED)
BRIDGE	10	APPID_SRCLEANING

CLI Keyword	Numeric Equivalent	Application ID
STP	11	APPID_SPANNINGTREE
LINKAGG	12	APPID_LINKAGGREGATION
QOS	13	APPID_QOS
RSVP	14	APPID_RSVP
IP	15	APPID_IP
IPMS	17	APPID_IPMS
AMAP	18	APPID_XMAP
GMAP	19	APPID_GMAP
AAA	20	APPID_AAA
IPC-MON	21	APPID_IPC_MON
IP-HELPER	22	APPID_BOOTP_RELAY
PMM	23	APPID_MIRRORING_MONITORING
MODULE	24	APPID_L3HRE
SLB	25	APPID_SLB
EIPC	26	APPID_EIPC
CHASSIS	64	APPID_CHASSISUPER
PORT-MGR	65	APPID_PORT_MANAGER
CONFIG	66	APPID_CONFIGMANAGER
CLI	67	APPID_CLI
SNMP	68	APPID_SNMP_AGENT
WEB	69	APPID_WEBMGT
MIPGW	70	APPID_MIPGW
SESSION	71	APPID_SESSION_MANAGER
TRAP	72	APPID_TRAP_MANAGER
POLICY	73	APPID_POLICY_MANAGER
DRC	74	APPID_DRC
SYSTEM	75	APPID_SYSTEM_SERVICES
HEALTH	76	APPID_HEALTHMON
NAN-DRIVER	78	APPID_NAN_DRIVER
RMON	79	APPID_RMON
TELNET	80	APPID_TELNET
PSM	81	APPID_PSM
FTP	82	APPID_FTP
SMNI	83	APPID_SMNI
DISTRIB	84	APPID_DISTRIB

CLI Keyword	Numeric Equivalent	Application ID
EPILOGUE	85	APPID_EPILOGUE
LDAP	86	APPID_LDAP
NOSNMP	87	APPID_NOSNMP
SSL	88	APPID_SSL
DBGGW	89	APPID_DBGGW
LANPOWER	108	APPID_LANPOWER

The **level** keyword assigns the error-type severity level to the specified application IDs. Values range from 2 (highest severity) to 9 (lowest severity). The values are defined in the following table:

Severity Level	Type	Description
2 (<i>highest severity</i>)	Alarm	A serious, non-recoverable error has occurred and the system must be rebooted.
3	Error	System functionality is reduced.
4	Alert	A violation has occurred.
5	Warning	An unexpected, non-critical event has occurred.
6 (<i>default</i>)	Info	Any other non-debug message.
7	Debug 1	A normal event debug message.
8	Debug 2	A debug-specific message.
9 (<i>lowest severity</i>)	Debug 3	A maximum verbosity debug message.

Specifying the Severity Level

To specify the switch logging severity level, use the **swlog appid level** command. The application ID can be expressed by using either the ID number or the application ID CLI keyword as listed in the table beginning on [page 28-6](#). The severity level can be expressed by using either the severity level number or the severity level type as shown in the table above. The following syntax assigns the “warning” severity level (or 5) to the “system” application, (ID number 75) by using the severity level and application names.

```
-> swlog appid system level warning
```

The following command makes the same assignment by using the severity level and application numbers.

```
-> swlog appid 75 level 3
```

No confirmation message appears on the screen for either command.

Removing the Severity Level

To remove the switch logging severity level, enter the **no swlog appid level** command, including the application ID and severity level values. The following is a typical example:

```
-> no swlog appid 75 level 5
```

Or, alternatively, as:

```
-> no swlog appid system level warning
```

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 socket ipaddr** 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.

Note. You can also send syslog files to multiple hosts (maximum of four).

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
```

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 clear** when the switch logging display is too long due to some of the data being old or out of date.

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 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!

A Software License and Copyright Statements

This appendix contains Alcatel-Lucent and third-party software vendor license and copyright statements.

Alcatel-Lucent License Agreement

ALCATEL-LUCENT 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 Alcatel-Lucent. Alcatel-Lucent 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 Alcatel-Lucent 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. **Alcatel-Lucent’s Rights.** Licensee acknowledges and agrees that the Licensed Materials are the sole property of Alcatel-Lucent 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 Alcatel-Lucent 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.** Alcatel-Lucent considers the Licensed Files to contain valuable trade secrets of Alcatel-Lucent, the unauthorized disclosure of which could cause irreparable harm to Alcatel-Lucent. 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 Alcatel-Lucent harmless from any claim, lawsuit, legal proceeding, settlement or judgment (including without limitation Alcatel-Lucent'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.** Alcatel-Lucent 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. Alcatel-Lucent 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 Alcatel-Lucent for either replacement or, if so elected by Alcatel-Lucent, 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 ALCATEL-LUCENT 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.** Alcatel-Lucent'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 Alcatel-Lucent for the Licensed Materials. IN NO EVENT SHALL ALCATEL-LUCENT BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, OR EXEMPLARY DAMAGES OR LOST PROFITS, EVEN IF ALCATEL-LUCENT 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 Alcatel-Lucent and Licensee, if any, Alcatel-Lucent is under no obligation to maintain or support the copies of the Licensed Files made and distributed hereunder and Alcatel-Lucent 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 Alcatel-Lucent and certifying to Alcatel-Lucent 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. Alcatel-Lucent may terminate this License Agreement upon the breach by Licensee of any term hereof. Upon such termination by

Alcatel-Lucent, Licensee agrees to return to Alcatel-Lucent 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 Alcatel-Lucent'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 Alcatel-Lucent by certain third party licensors. Some third party licensors are third party beneficiaries to this License Agreement with full rights of enforcement. Please refer to the section entitled "[Third Party Licenses and Notices](#)" on page A-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 into this release at the following URL: <https://service.esd.alcatel-lucent.com/portal/page/portal/EService/release>

Index

Numerics

- 10/100/1000 ports
 - defaults 1-3
- 802.1AB 10-1
 - defaults 10-2
 - specifications 10-2
 - verify information about 10-14
- 802.1Q
 - enabling notification 10-10
 - trusted ports 21-26
- 802.3ad
 - see* dynamic link aggregation

A

- aaa ldap-server** command
 - LDAP authentication 23-25
- aaa radius-server** command
 - RADIUS authentication 23-11, 23-13
- Access Control Lists
 - see* ACLs
- access list 15-15
 - creating 15-15
- ACLs
 - interaction with VRRP 18-10, 18-19
 - Layer 2 21-59
 - Layer 3 21-60
 - Layer 3 application examples 21-61
 - multicast 21-62
 - security features 21-63
- actions
 - combined with conditions 21-28, 21-29
 - creating policy actions 21-42
- Address Resolution Protocol
 - see* ARP
- alerts 28-8
- Application example
 - Learned Port Security Configuration 25-3
- application example
 - VLAN Stacking 11-2, 11-35
- application examples
 - authentication servers 23-4
 - Configuring 802.1AB 10-2
 - DHCP Relay 17-4, 17-7, 17-8
 - dynamic link aggregation 7-4, 7-27, 8-5
 - ICMP policies 21-72
 - IP 11-4
 - IPMS 20-36, 20-38
 - IPv6 13-4
 - Layer 3 ACLs 21-61
 - policies 21-69

- policy map groups 21-56
- Port Mapping 24-2, 24-6
- port mirroring 26-4
- port monitoring 26-6, 26-8
- QoS 21-39, 21-69
- RIP 15-3
- RMON 26-11
- Server Load Balancing 19-4
- Spanning Tree Algorithm and Protocol 5-10, 5-44
- static link aggregation 6-3, 6-10
- switch health 26-13
- switch logging 28-4
- UDLD 2-3
- VLANs 4-3, 4-11
- VRRP 18-5, 18-26, 18-30
- VRRP3 18-31
- applied configuration 21-66
 - how to verify 21-68
- ARP
 - clearing the ARP cache 11-13
 - creating a permanent entry 11-12
 - deleting a permanent entry 11-13
 - dynamic entry 11-12
 - filtering 11-14
 - local proxy 11-14
- arp** command 11-12
- arp filter** command 11-14
- assigning ports to VLANs 4-7
- Authenticated Switch Access
 - LDAP VSAs 23-20
- authentication servers
 - application example 23-4
 - defaults 23-3
 - how backups work 23-5
 - see* LDAP authentication servers, RADIUS authentication servers
- automatic IP configuration 17-12

B

- backup router
 - VRRP 18-7
- BGP IPv6
 - configuring 11-35
- boundary port 5-18
- BPDU
 - see* Bridge Protocol Data Units
- bridge 1x1 slot/port path cost** command 5-37
- bridge auto-vlan-containment** command 5-32
- bridge cist hello time** command 5-27, 5-28, 5-29, 5-30
- bridge forward delay** command 5-30
- bridge hello time** command 5-28
- bridge max age** command 5-29, 5-31
- bridge mode** command 5-20
- bridge msti priority command** 5-28
- bridge path cost mode** command 5-31
- bridge priority** command 5-28
- Bridge Protocol Data Units
 - contents 5-8

- bridge slot/port** command 5-30
 - bridge slot/port connection** command 5-41
 - bridge slot/port priority** command 5-36
 - built-in port groups
 - used with Policy Based Routing 21-74
- C**
- clear arp filter** command 11-15
 - clear arp-cache** command 11-13
 - condition groups
 - for ACLs 21-48
 - MAC groups 21-52
 - network groups 21-49
 - port groups 21-53
 - sample configuration 21-48
 - service groups 21-51
 - verify information about 21-55
 - conditions
 - combined with actions 21-28, 21-29
 - configuring 21-41
 - how to create 21-41
 - see also* condition groups
 - valid combinations 21-27
 - Configuring 802.1AB
 - application examples 10-2
- D**
- debug messages 28-8
 - debug qos** command 21-35
 - default route
 - IP 11-12
 - defaults
 - 10/100/1000 ports 1-3
 - 802.1AB 10-2
 - authentication servers 23-3
 - DHCP Relay 17-3
 - dynamic link aggregation 7-3, 8-4
 - Ethernet ports 1-2, 1-3
 - IP 11-3
 - IPMS 20-3, 20-4
 - IPv6 13-3
 - Learned Port Security 25-2
 - Multiple Spanning Tree 5-5
 - OSPF 14-2, 16-2
 - policy servers 22-2
 - Port Mapping 24-2
 - port mirroring 26-3
 - port monitoring 26-5, 26-7
 - QoS 21-30
 - RIP 15-2
 - RMON 26-11
 - RRSTP 5-6
 - Server Load Balancing 19-3
 - source learning 3-2
 - Spanning Tree Bridge 5-4, 9-2
 - Spanning Tree Port 5-4
 - static link aggregation 6-2
 - switch health 26-13
 - switch logging 28-3
 - UDLD 2-2
 - VLANs 4-2
 - VRRP 18-3
 - Denial of Service
 - see* DoS
 - DHCP 17-6
 - DHCP Relay 17-1, 17-10
 - application examples 17-4, 17-7, 17-8
 - AVLAN forwarding option 17-11
 - defaults 17-3
 - DHCP server IP address 17-9
 - forward delay time 17-10
 - maximum number of hops 17-11
 - standard forwarding option 17-11
 - directed broadcast 11-23
 - DoS 11-23
 - enabling traps 11-27
 - setting decay value 11-27
 - setting penalty values 11-26
 - Setting Port Scan Penalty Value 11-26
 - DVMRP 20-7
 - dynamic link aggregation 7-1, 8-1
 - application examples 7-4, 7-27, 8-5
 - defaults 7-3, 8-4
 - group actor administrative key 7-14
 - group actor system ID 7-15
 - group actor system priority 7-14
 - group administrative state 7-13
 - group partner administrative key 7-15
 - group partner system ID 7-16
 - group partner system priority 7-16
 - groups 7-9
 - assigning ports 7-10
 - creating groups 7-9
 - deleting groups 7-10
 - group names 7-13
 - removing ports 7-11
 - LACPDU bit settings 7-17, 7-21
 - LACPDU frames 7-17, 7-21
 - Link Aggregation Control Protocol (LACP) 7-6
 - MAC address 7-15, 7-16, 7-19, 7-23
 - port actor administrative priority 7-19
 - port actor port priority 7-20
 - port actor system administrative states 7-17
 - port actor system ID 7-19
 - port partner administrative key 7-23
 - port partner administrative priority 7-25
 - port partner administrative state 7-21
 - port partner administrative system ID 7-23
 - port partner administrative system priority 7-24
 - port partner port administrative status 7-25
 - ports 7-10
 - remote group MAC address 8-30
 - specifications 7-2, 8-3
 - verify information about 7-30
 - dynamic log
 - LDAP accounting servers 23-24

E

errors 28-8
Ethernet
 defaults 1-2, 1-3
 flood rate 1-6
 frame size 1-5
 full duplex 1-4
 half duplex 1-4
 specifications 1-2

F

Fast Spanning Tree 5-6
filtering lists
 see ACLs
flow command 1-6

H

health interval command 26-45
health threshold command 26-43
health threshold limits
 displaying 26-44
Hot Standby Routing Protocol
 see HSRP
HSRP
 not compatible with VRRP 18-3

I

ICMP 11-29
 control 11-32
 QoS policies for 21-72
 statistics 11-32
icmp messages command 11-30
icmp type command 11-30
IGMP
 multicast ACLs 21-59, 21-62
IGMP Spoofing 20-18
interfaces admin command 1-5
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 1-4
Internet Control Message Protocol
 see ICMP
IP 11-1, 12-1
 application examples 11-4, 12-3
 ARP 11-12
 defaults 11-3
 directed broadcast 11-23
 ICMP 11-29
 ping 11-32
 protocols 11-5, 12-9
 router ID 11-16
 router port 11-7

 router primary address 11-16
 specifications 11-3
 static route 11-11, 13-18
 tracing an IP route 11-33
 TTL value 11-17
 UDP 11-33
 verify information about 11-37, 12-15
ip access-list address command 15-15
ip access-list command 15-15
ip default-ttl command 11-17
ip directed-broadcast command 11-23
ip dos scan close-port-penalty command 11-26
ip dos scan decay command 11-27
ip dos scan tcp open-port-penalty command 11-26
ip dos scan threshold command 11-26
ip dos scan udp open-port-penalty command 11-26
ip dos trap command 11-27
ip helper address command 17-9
ip helper boot-up command 17-12
ip helper forward delay command 17-10
ip helper maximum hops command 17-11
ip helper per-vlan command 17-11
ip helper standard command 17-11
ip interface command 15-3
ip load rip command 15-3, 15-6
ip multicast igmp-proxy-version command 20-9, 20-24
ip multicast neighbor-timeout command 20-9, 20-15, 20-16, 20-17, 20-24, 20-31
ip multicast query-interval command 20-13, 20-14, 20-28
ip multicast static-member command 20-11
ip multicast static-neighbor command 20-25
ip multicast static-querier command 20-11
IP Multicast Switching
 see IPMS
ip multicast switching command 20-8, 20-18, 20-23, 20-33
IP multinetting 11-7
ip redist command 15-12
ip rip force-holddowntimer command 15-9
ip rip garbage-timer command 15-10
ip rip holddown-timer command 15-10
ip rip host-route command 15-11
ip rip interface auth-key command 15-18
ip rip interface auth-type command 15-18
ip rip interface command 15-3, 15-7
ip rip interface metric command 15-8
ip rip interface recv-version command 15-8
ip rip interface send-version command 15-7
ip rip interface status command 15-3, 15-7
ip rip invalid-timer command 15-10
ip rip route-tag command 15-9
ip rip status command 15-3, 15-7
ip rip update-interval command 15-9
ip route-pref command 11-16
IP router ports 11-8
 modifying 11-9
 removing 11-9, 12-14
ip router primary-address command 11-16
ip router router-id command 11-16
ip service command 11-28

ip slb admin command 19-4, 19-11
ip slb cluster admin status command 19-17
ip slb cluster command 19-4, 19-5, 19-12
ip slb cluster ping period command 19-15
ip slb cluster ping retries command 19-16
ip slb cluster ping timeout command 19-15
ip slb probe command 19-18, 19-19
ip slb probe expect command 19-21
ip slb probe password command 19-20
ip slb probe period command 19-19
ip slb probe port command 19-20
ip slb probe retries command 19-20
ip slb probe send command 19-21
ip slb probe status command 19-21
ip slb probe timeout command 19-19
ip slb probe url command 19-20
ip slb probe username command 19-20
ip slb server ip cluster command 19-4, 19-5, 19-14, 19-16, 19-17
ip static-route command 11-11, 13-18
IPMS 20-1

- adding static members 20-12
- adding static neighbors 20-10
- adding static queriers 20-11
- application examples 20-36, 20-38
- defaults 20-3, 20-4
- deleting static members 20-12, 20-27
- deleting static neighbors 20-11
- deleting static queriers 20-11, 20-26
- displaying 20-40, 20-41
- DVMRP 20-7
- enabling 20-8, 20-18, 20-19, 20-20, 20-33, 20-34
- IGMPv2 20-10, 20-25
- IGMPv3 20-7, 20-10, 20-24
- neighbor timeout 20-15, 20-16, 20-17, 20-30, 20-32
- optional multicast routing software 20-6
- overview 20-5
- PIM-SM 20-7
- query interval 20-13, 20-14, 20-28, 20-29
- RFCs 20-2
- specifications 20-2

IPv6 13-1

- addressing 13-6
- application examples 13-4
- autoconfiguration of addresses 13-8
- defaults 13-3
- specification 13-2
- tunneling types 13-17
- verify information about 13-26

ipv6 access-list address command 15-15
ipv6 access-list command 15-15
ipv6 address command 13-4, 13-15
ipv6 interface command 13-4, 13-13, 13-14
ipv6 interface tunnel source destination command 13-13
ipv6 load rip command 13-4
ipv6 rip interface command 13-4
ipv6 route-pref command 13-19

J

jumbo frames 1-2

L

LACP

see dynamic link aggregation

lacp agg actor admin key command 7-4, 7-10
lacp agg actor admin state command 7-17
lacp agg actor port priority command 7-20
lacp agg actor system id command 7-19
lacp agg actor system priority command 7-19
lacp agg partner admin key command 7-23
lacp agg partner admin port command 7-25
lacp agg partner admin port priority command 7-25
lacp agg partner admin state command 7-21
lacp agg partner admin system id command 7-23
lacp agg partner admin system priority command 7-24
lacp linkagg actor admin key command 7-14
lacp linkagg actor system id command 7-15
lacp linkagg actor system priority command 7-14
lacp linkagg admin state command 7-13
lacp linkagg name command 7-13
lacp linkagg partner admin key command 7-15
lacp linkagg partner system id command 7-16
lacp linkagg partner system priority command 7-16
lacp linkagg size command 7-4, 7-9, 8-5, 8-7, 8-23, 8-24

Layer 2

statistics counters 1-4

LDAP accounting servers

dynamic log 23-24
standard attributes 23-22

LDAP authentication servers

directory entries 23-16
functional privileges 23-21
passwords for 23-19
schema extensions 23-16
SNMP attributes on authentication servers 23-21
SSL 23-26
VSAs for Authenticated Switch Access 23-20

LDAP servers

see policy servers
used for QoS policies 22-3

Learned Port Security

database table 25-7
defaults 25-2
disabling 25-9
enabling 25-9
overview 25-5
specifications 25-2

Learned Port Security Configuration

Application example 25-3

Lightweight Directory Access Protocol

see LDAP servers

line speed 1-4

link aggregation

dynamic link aggregation 7-1, 8-1
Spanning Tree parameters 5-35, 5-36, 5-38, 5-40, 5-42
static link aggregation 6-1

lldp lldpdu command 10-2
lldp notification command 10-2
lldp tlv dot1 command 10-11
lldp tlv dot3 command 10-11
lldp tlv management command 10-3
lldp tlv med command 10-12
logged events
 detail level 21-36
 types of events 21-35

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, 5-29
 dynamic link aggregation 7-15, 7-16, 7-19, 7-23, 8-30
 learned 3-3
 statically assigned 3-3
mac-address-table command 3-3
map groups 21-56
 application 21-73
 creating 21-57
 verifying information 21-58
master router
 VRRP 18-7
MLD Zapping 20-33
MST 5-12
 Internal Spanning Tree (IST) Instance 5-17
 Interoperability 5-18
 Migration 5-18, 5-19
 MSTI 5-15
 MSTP 5-12
 Multiple Spanning Tree Region 5-16
Multicast Listener Discovery (MLD) 20-24
Multiple Spanning Tree
 defaults 5-5

N

non combo ports
 configuring 1-3

O

OSPF 15-4
 defaults 14-2, 16-2
 loading software 16-11
 specifications 14-2, 16-2
OSPF redistribution policies
 deleting 11-19, 11-21, 13-22, 13-24, 15-16

P

pending configuration 21-66
pending policies
 deleting 21-66
 Per VLAN DHCP 17-9
 PIM-SM 20-7
 ping
 IP 11-32
ping command 11-32
policies
 application examples 21-69
 applied 21-66
 built-in 21-33
 conditions 21-41
 creating policy actions 21-42
 how the switch uses them 21-26
 Policy Based Routing 21-74
 precedence 21-45
 redirect linkagg 21-71
 redirect port 21-71
 rules 21-43
 verify information about 21-47
policies configured via PolicyView 21-67
policy action 802.1p command 21-9
policy action command 21-39
policy action map command 21-56
policy action redirect linkagg command 21-71
policy action redirect port command 21-71, 21-72
policy actions
 see actions
Policy Based Routing 21-74
policy condition command 21-39
policy conditions
 see conditions
policy MAC groups 21-52
policy map group command 21-56
policy map groups
 application example 21-56
policy network group command 21-48
policy network groups 21-49
 switch default group 21-33, 21-49
policy port group command 21-48
policy port groups 21-53
policy rule command 21-39
policy server command 22-2, 22-4
policy server flush command 22-6
 compared to **qos flush** command 22-7
policy server load command 22-6
policy servers
 defaults 22-2
 downloading policies 22-6
 installing 22-3
 SSL 22-6
policy service group command 21-48
policy service groups 21-51
policy services 21-50
PolicyView
 LDAP policy servers 22-1
Port Mapping 24-1
 application examples 24-2, 24-6
 defaults 24-2
 specifications 24-2
port mapping command 24-2

-
- Port Mapping Session
 - creating and deleting 24-3
 - enabling and disabling 24-4
 - port mirroring 26-14
 - application examples 26-4
 - defaults 26-3
 - direction 26-20
 - disabling mirroring status 26-19
 - displaying status 26-21
 - enabling or disabling mirroring status 26-19
 - N-to-1 port mirroring 26-18
 - specifications 26-3
 - unblocking ports 26-19
 - port mirroring** command 26-20
 - port mirroring session
 - creating 26-18
 - deleting 26-21
 - enabling/disabling 26-20
 - port mirroring source** command 26-6
 - port mirroring source destination** command 26-18, 26-19, 26-20
 - port monitoring
 - application examples 26-6, 26-8
 - configuring 26-24, 26-30, 26-31
 - creating a data file 26-25
 - defaults 26-5, 26-7
 - deleting a session 26-25, 26-33
 - direction 26-26
 - disabling a session 26-25
 - displaying status and data 26-27, 26-31, 26-33
 - enabling a session 26-24
 - file overwriting 26-26
 - file size 26-26
 - overview 26-23, 26-29
 - pausing a session 26-25
 - resuming a session 26-25
 - session persistence 26-25
 - specifications 26-5, 26-7
 - port monitoring** command 26-25
 - port monitoring source** command 26-24, 26-25, 26-26, 26-30
 - ports
 - displaying QoS information about 21-9
 - Spanning Tree parameters 5-33
 - VLAN assignment 4-7
 - port-security** command 25-9
 - port-security shutdown** command 25-10
 - Precedence
 - Configured rule order 21-45
 - Precedence value 21-45
 - precedence
 - for policies 21-45
- Q**
- QoS
- application examples 21-39, 21-69
 - ASCII-file-only syntax 12-8, 21-40
 - configuration overview 21-34
 - defaults 21-30
 - enabled/disabled 21-35
 - interaction with other features 21-26
 - overview 21-4
 - quick steps for creating policies 21-39
 - Server Load Balancing 19-13
 - Specifications 21-3
 - traffic prioritization 21-69
 - qos apply** command 21-66
 - global configuration 21-66
 - policy and port configuration 21-66
 - qos clear log** command 21-38
 - qos** command 21-35
 - qos flush** command 21-67
 - compared to **policy server flush** command 22-7
 - qos forward log** command 21-36
 - QoS log
 - cleared 21-38
 - displayed 21-37
 - number of display lines 21-36
 - see also* logged events
 - qos log level** command 21-35, 21-36
 - qos port default 802.1p** command 21-8
 - qos port default dscp** command 21-8
 - qos port trusted** command 21-9
 - qos reset** command 21-38
 - qos revert** command 21-66
 - qos trust ports** command 21-8
 - qos user-port** command 21-64
 - Quality of Service
 - see* QoS
- R**
- RADIUS accounting servers
 - standard attributes 23-10
 - VSAs 23-11
 - RADIUS authentication servers 23-7
 - standard attributes 23-7
 - VSAs 23-9
 - Rapid Spanning Tree Algorithm and Protocol
 - see* RSTP
 - Redirection Policies 21-71
 - Remote Authentication Dial-In User Service
 - see* RADIUS authentication servers
 - resource threshold limits
 - configuring 26-43
 - Ring Rapid Spanning Tree Algorithm and Protocol
 - see* RRSTP
 - RIP 15-1
 - application examples 15-3
 - defaults 15-2
 - enabling 15-7
 - forced hold-down timer 15-9
 - garbage timer 15-10
 - hold-down timer 15-10
 - host route 15-11
 - interface 15-7
 - invalid timer 15-10

- IP 15-4
 - loading 15-6
 - redistribution 15-12
 - security 15-18
 - specifications 15-2
 - unloading 15-6
 - update interval 15-9
 - verification 15-19
 - verify information about 15-19
 - RIP interface
 - creating 15-7
 - deleting 15-7
 - enabling 15-7
 - metric 15-8
 - password 15-18
 - receive option 15-8
 - route tag 15-9
 - send option 15-7
 - RMON
 - application examples 26-11
 - defaults 26-11
 - specifications 26-10
 - RMON events
 - displaying list 26-40
 - displaying specific 26-40
 - RMON probes
 - displaying list 26-37
 - displaying statistics 26-38
 - enabling/disabling 26-36
 - rmon probes** command 26-36
 - RMON tables
 - displaying 26-37
 - round robin distribution algorithm
 - see* weighted round robin distribution algorithm
 - route map
 - creating 15-13
 - deleting 15-14
 - enabling/disabling administrative status 15-16
 - redistribution 15-16
 - sequencing 15-14
 - router ID 11-16, 13-19
 - router port
 - IP 11-7
 - router primary address 11-16
 - Routing Information Protocol
 - see* RIP
 - RRSTP 5-44
 - defaults 5-6
 - RSTP 5-6
 - port connection types 5-41
 - rules
 - see* policies
- S**
- sampling intervals
 - configuring 26-45
 - viewing 26-45
 - Secure Socket Layer
 - see* SSL
 - Security Violation Mode 25-12
 - restrict** mode 25-12
 - shutdown** mode 25-13
 - server clusters 19-12, 19-17
 - server distribution algorithms 19-9
 - Server Load Balancing 19-1
 - adding servers 19-14
 - application examples 19-4
 - clusters 19-12, 19-17
 - configuration steps 19-11
 - defaults 19-3
 - deleting clusters 19-14
 - deleting servers 19-14
 - disabling 19-11
 - disabling clusters 19-17
 - disabling servers 19-18
 - displaying 19-22
 - distribution algorithms 19-9
 - enabling 19-11
 - enabling clusters 19-17
 - enabling servers 19-17
 - ping period 19-15
 - ping retries 19-16
 - ping timeout 19-15
 - QoS 19-13
 - relative server weight 19-16
 - server health monitoring 19-10
 - servers 19-14, 19-17
 - specifications 19-2
 - weighted round robin distribution algorithm 19-9
 - Server Load Balancing probes 19-18
 - clusters 19-19
 - configuring 19-18
 - deleting 19-18
 - expected status 19-21
 - modifying 19-19
 - password 19-20
 - probe expect 19-21
 - probe send 19-21
 - retries 19-20
 - servers 19-19
 - TCP/UDP port 19-20
 - timeout 19-19
 - URL 19-20
 - user name 19-20
 - severity level
 - see* switch logging
 - show arp** command 11-13
 - show arp filter** command 11-15, 11-27
 - show health** command 26-46
 - show health interval** command 26-45
 - show health threshold** command 26-13, 26-44
 - show icmp control** command 11-32
 - show icmp statistics** command 11-32
 - show ip config** command 11-17, 11-23
 - show ip interface** command 11-9
 - show ip redist** command 15-16

show ip rip command 15-7
show ip rip interface command 15-7
show ip route command 11-11, 13-18
show ip route-map command 15-13
show ipv6 interface command 13-14
show linkagg command 6-11
show linkagg port command 6-11
show lldp remote-system command 10-4
show lldp statistics command 10-3
show log swlog command 28-12
show policy rule command 19-13
show policy server long command 22-6
show port mirroring status command 26-21
show port monitoring file command 26-27
show port-security command 25-3
show port-security shutdown command 25-4
show qos log command 21-37
show rmon events command 26-37
show rmon probes command 26-11, 26-37
show tcp ports command 11-33
show tcp statistics command 11-33
show udld configuration command 2-3
show udld statistics port command 2-3
show udp ports command 11-33
show udp statistics command 11-33
show vlan svlan command 8-30, 27-24
show vlan svlan port-config command 8-30, 27-24
SLB
see Server Load Balancing
SNMP
attributes for LDAP authentication servers 23-21
source learning 3-1
defaults 3-2
MAC address table 3-1, 3-3
source learning time limit 25-10
Spanning Tree
specifications 5-3, 9-2
Spanning Tree Algorithm and Protocol 5-1, 9-1
1x1 operating mode 4-9, 5-20, 5-22
application examples 5-10, 5-44
bridge ID 5-8, 5-27
Bridge Protocol Data Units 5-8, 5-28, 5-29
bridged ports 5-33
designated bridge 5-6
flat operating mode 4-9, 5-20, 5-21
path cost 5-37
port connection types 5-41
Port ID 5-8
port ID 5-36
port path cost 5-6
port roles 5-6
port states 5-7, 5-40
root bridge 5-6, 5-28, 5-29
root path cost 5-6
topology 5-6, 5-11
Topology Change Notification 5-9
Spanning Tree Bridge
defaults 5-4, 9-2
Spanning Tree bridge parameters
802.1D standard protocol 5-27
802.1s multiple spanning tree protocol 5-27
802.1w rapid reconfiguration protocol 5-27
automatic VLAN containment 5-31
forward delay time 5-29
hello time 5-28
maximum age time 5-29
priority 5-27
Spanning Tree Port
defaults 5-4
Spanning Tree port parameters 5-33
connection type 5-41
link aggregate ports 5-35, 5-36, 5-38, 5-40, 5-42
mode 5-40
path cost 5-37
priority 5-36
specification
IPv6 13-2
Specifications
QoS 21-3
specifications
802.1AB 10-2
dynamic link aggregation 7-2, 8-3
Ethernet 1-2
IP 11-3
OSPF 14-2, 16-2
Port Mapping 24-2
port mirroring 26-3
port monitoring 26-5, 26-7
RIP 15-2
RMON 26-10
Server Load Balancing 19-2
Spanning Tree 5-3, 9-2
static link aggregation 6-2
switch health 26-12
switch logging 28-2
UDLD 2-2
SSL
for LDAP authentication servers 23-26
policy servers 22-6
static agg agg num command 6-3, 6-8
static link aggregation 6-1
adding ports 6-8
application examples 6-3, 6-10
configuration steps 6-6
creating 6-7
defaults 6-2
deleting 6-8
deleting ports 6-8
disabling 6-9
enabling 6-9
group names 6-9
groups 6-5, 7-6
overview 6-5, 7-6
specifications 6-2
verify information about 6-11
static linkagg admin state command 6-9
static linkagg name command 6-9

static linkagg size command 6-3, 6-7
 static MAC addresses 3-3
 static route
 IP 11-11, 13-18
 metric 11-11, 13-18
 subnet mask 11-11
 subnet mask 11-11
 switch health
 application examples 26-13
 defaults 26-13
 monitoring 26-41
 specifications 26-12
 switch health statistics
 resetting 26-47
 viewing 26-46
 switch logging
 application examples 28-4
 application ID 28-6
 defaults 28-3
 output 28-9
 severity level 28-8
 specifications 28-2
swlog appid level command 28-6
swlog clear command 28-11
swlog command 28-4, 28-6
swlog output command 21-37
swlog output command 28-9

T

TCN BPDU
 see Topology Change Notification BPDU
 TCP
 statistics 11-33
 time-to-live
 see TTL
 Topology Change Notification BPDU 5-9
traceroute command 11-33
 tracking
 VRRP 18-9
 traffic prioritization 21-69
trap port link command 1-4
 traps
 port link messages 1-4
 trusted ports
 see also ports
 used with QoS policies 21-9
 TTL value 11-17
 Tunneling 13-10

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
 specifications 2-2
udld command 2-3
udld port command 2-3
 UDP 11-33
 statistics 11-33
 User Datagram Protocol
 see UDP
 users
 functional privileges 23-21

V

Vendor Specific Attributes
 see VSAs
 Virtual Router Redundancy Protocol
 see VRRP
 virtual routers 18-7
vlan 802.1q command 4-6
vlan authentication command 4-9
vlan command 11-4, 12-3, 15-3
vlan port default command 4-6, 4-7, 11-4, 15-3
vlan router ip command 11-4, 12-3, 12-4
 VLAN Stacking
 application example 11-2, 11-35
 display list of all or range of configured SVLANs 9-19
 displaying the configuration 27-24
vlan stp command 4-9
vlan svlan command 5-26
 VLANs 4-1, 4-5, 9-3
 administrative status 4-6
 application examples 4-3, 4-11
 default VLAN 4-7
 defaults 4-2
 description 4-6
 IP multinetting 11-7
 IP router ports 11-8
 MAC address aging time 3-7
 operational status 4-5
 port assignment 4-7
 Spanning Tree status 4-9
 VLAN ID 4-4
 VRRP 18-1
 ACLs 18-10, 18-19
 application example 18-5, 18-26, 18-30
 ARP request 18-8
 backup router 18-7
 defaults 18-3
 MAC address 18-8
 master router 18-7
 tracking 18-9
 virtual routers 18-7
vrrp command 18-10, 18-19
 defaults 18-3
vrrp delay command 18-14
vrrp ip command 18-10, 18-19
vrrp track command 18-25
vrrp track-association command 18-25
vrrp trap command 18-14, 18-23

VRRP3 18-19
Advertisement Interval 18-21
application examples 18-31
Preemption 18-22
Traps 18-23
Virtual Router 18-19
Virtual Router Priority 18-21

VSAs
for LDAP servers 23-20
for RADIUS authentication 23-7
RADIUS accounting servers 23-11
setting up for RADIUS servers 23-9

W

warnings 28-8
weighted round robin distribution algorithm 19-9