OmniSwitch AOS Release 8 Data Center Switching Guide

8.4.1.R03

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

Alcatel·Lucent
Enterprise

www.al-enterprise.com

**This user guide documents AOS Release 8.4.1.R03 for the
OmniSwitch 9900, OmniSwitch 6900, OmniSwitch 6860, and OmniSwitch 6865.
The functionality described in this guide is subject to change without notice.**

Alcatel·Lucent
Enterprise

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

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

Contents

About This Guide

This *OmniSwitch AOS Release 8 Data Center Switching Guide* describes how to set up and monitor supported protocols and software features that comprise the Alcatel-Lucent Enterprise data center switching architecture. Some of the features described in this guide are purchased as an add-on package to the base switch software.

Supported Platforms

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

- OmniSwitch 9900 Series

- OmniSwitch 6900 Series

- OmniSwitch 6860 Series

- OmniSwitch 6865 Series

Who Should Read this Manual?

The audience for this user guide are network administrators and IT support personnel who need to configure, maintain, and monitor switches and routers in a live network. However, anyone wishing to gain knowledge about how software features supporting data center switching functionality 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 you are ready to integrate your OmniSwitch into your network and you are ready to set up the data center switching protocols and features. You should already be familiar with the basics of managing a single OmniSwitch as described in the *OmniSwitch AOS Release 8 Switch Management Guide*.

The topics and procedures in this manual assume an understanding of the OmniSwitch directory structure and basic switch administration commands and procedures.

What is in this Manual?

This switching guide includes a "Data Center Switching Introduction" chapter that provides a description of the OmniSwitch data center switching architecture and software features. In addition to this introduction, this guide also includes information about configuring the following features;

* Data Center Bridging (DCB) protocols.

* Shortest Path Bridging MAC (SPBM), including SPBM support of Provider Backbone Bridging (PBB) encapsulation and services.

* Universal Network Profile (UNP), specifically for configuring Virtual Network Profiles (vNP) to manage virtual machines within and between data centers.

* Edge Virtual Bridging (EVB) for managing virtual machines created and managed on servers also running the EVB protocol.

* Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) snooping to ensure the security of an FCoE network.

* FCoE/FC gateway functionality to converge FC over Ethernet and FC-to-FC over Ethernet through an OmniSwitch gateway.

* Virtual eXtensible Local Area Network (VXLAN) to transparently extend Layer 2 networks over a Layer 3 infrastructure.

* VXLAN Snooping to detect and identify VXLAN traffic on the network.

What is Not in this Manual?

The configuration procedures in this manual use Command Line Interface (CLI) commands in all examples. CLI commands are text-based commands used to manage the switch through serial (console port) connections or via Telnet sessions. Procedures for other switch management methods, such as web-based (WebView or OmniVista) or SNMP, are outside the scope of this guide.

For information on WebView and SNMP switch management methods consult the *OmniSwitch AOS Release 8 Switch Management Guide.* Information on using WebView and OmniVista can be found in the context-sensitive on-line help available with those network management applications.

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

How is the Information Organized?

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

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

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

Documentation Roadmap

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

Stage 1: Using the Switch for the First Time

Pertinent Documentation: *OmniSwitch Hardware Users Guide*
                                        Release Notes

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

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

Stage 2: Gaining Familiarity with Basic Switch Functions

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

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

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

Stage 3: Integrating the Switch Into a Network

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

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

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

The *OmniSwitch AOS Release 8 Data Center Switching Guide* includes configuration information for data center networks using virtualization technologies (SPBM and UNP), Data Center Bridging protocols (PFC, ETC, and DCBX), and FCoE/FC gateway functionality.

Anytime

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

Related Documentation

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

- *OmniSwitch 9900, 6900, 6860, 6865 Hardware Users Guides*

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

- *OmniSwitch AOS Release 8 CLI Reference Guide*

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

- *OmniSwitch AOS Release 8 Switch Management Guide*

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

- *OmniSwitch AOS Release 8 Network Configuration Guide*

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

- *OmniSwitch AOS Release 8 Advanced Routing Configuration Guide*

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

- *OmniSwitch AOS Release 8 Data Center Switching Guide*

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

- *OmniSwitch AOS Release 8 Transceivers Guide*

  Includes SFP and XFP transceiver specifications and product compatibility information.

- *OmniSwitch AOS Release 8 Specifications Guide*

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

- Technical Tips, Field Notices

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

- *Release Notes*

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

Technical Support

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

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

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

> Web: support.esd.alcatel-lucent.com

> Phone: 1-800-995-2696

> Email: ebg_global_supportcenter@al-enterprise.com

1 Understanding Data Center Switching

Alcatel-Lucent Enterprise helps enterprises address the challenges and ongoing transformation of data center networks while delivering a high-quality user experience for new real-time applications, greater agility in deploying new applications, seamless integration of public cloud services, and reduced data center costs. To deliver on these capabilities, Alcatel-Lucent Enterprise provides a unique blueprint for data center switching that brings together three core innovations:

- **OmniSwitch Pod**. A unique architecture concept for top-of-rack switches that can provide server-to-server connectivity without the need for a core switch to carry traffic. The pod is a highly dense architecture that ensures low latency and high performance between servers connected to the same pod.

- **Data Center Mesh**. The mesh consists of pods connected to each other and to core switches to bring together thousands of server-facing ports with low aggregate end-to-end latency. Implementing the data center mesh architecture allows enterprises to create virtual data centers supporting defined workgroups or departments.

- **Virtual Network Profile (vNP).** A type of Universal Network Profile (UNP) that the administrator configures specifically for virtual machine classification. A virtual machine is bound to a vNP to ensure consistent application of network access controls and QoS policies when a virtual machine is initially detected or moves to a different network location.

In This Chapter

This chapter contains an overview of the OmniSwitch components and features of the Alcatel-Lucent Enterprise Data Center Switching solution. It provides a general example for configuring the related data center software applications through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following topics are included in this chapter:

Data Center Switching Components

Key components of the OmniSwitch data center switching framework include an OmniSwitch pod, the data center mesh, and virtual network profiles (vNP). The OmniSwitch also supports the use of other network virtualization technology features, such as Shortest Path Bridging (SPB), Data Center Bridging (DCB), and Virtual Chassis.

OmniSwitch Pod

The OmniSwitch pod is an architecture concept for top-of-rack switches that provides server-to-server (east-west and north-south) connectivity without the need for a core switch to carry traffic. The pod is a highly dense architecture that ensures low latency and high performance between servers connected to the same pod.

The architecture of the OmniSwitch pod provides the scalability necessary to handle changing data center demands. A pod may initially only consist of one or two such switches, but as the demand to handle more and more traffic grows, additional switches are easily added into the pod configuration.

Figure 1 shows some examples of OmniSwitch pod architectures.
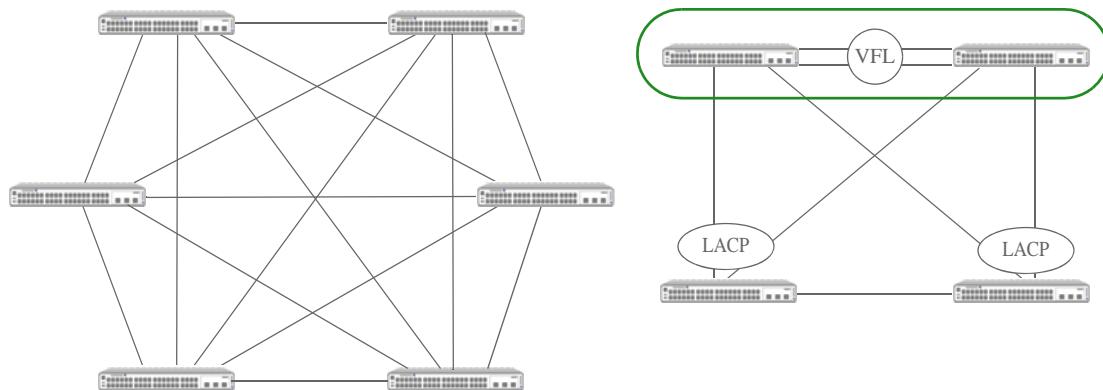
Figure 1: OmniSwitch Pod Examples

Each pod is a switching fabric where every switch could be connected to every other switch. When pods are then connected together with other pods and core switches, a data center mesh is formed.

Data Center Mesh

The data center mesh consists of pods connected to each other and to core switches to bring together thousands of server-facing ports with low aggregate end-to-end latency and high resiliency. Implementing the data center mesh architecture allows enterprises to create virtual data centers supporting virtualized workgroups or departments. Figure 2 shows an example of a data center mesh that consists of four OmniSwitch pods and two OmniSwitch core switches:
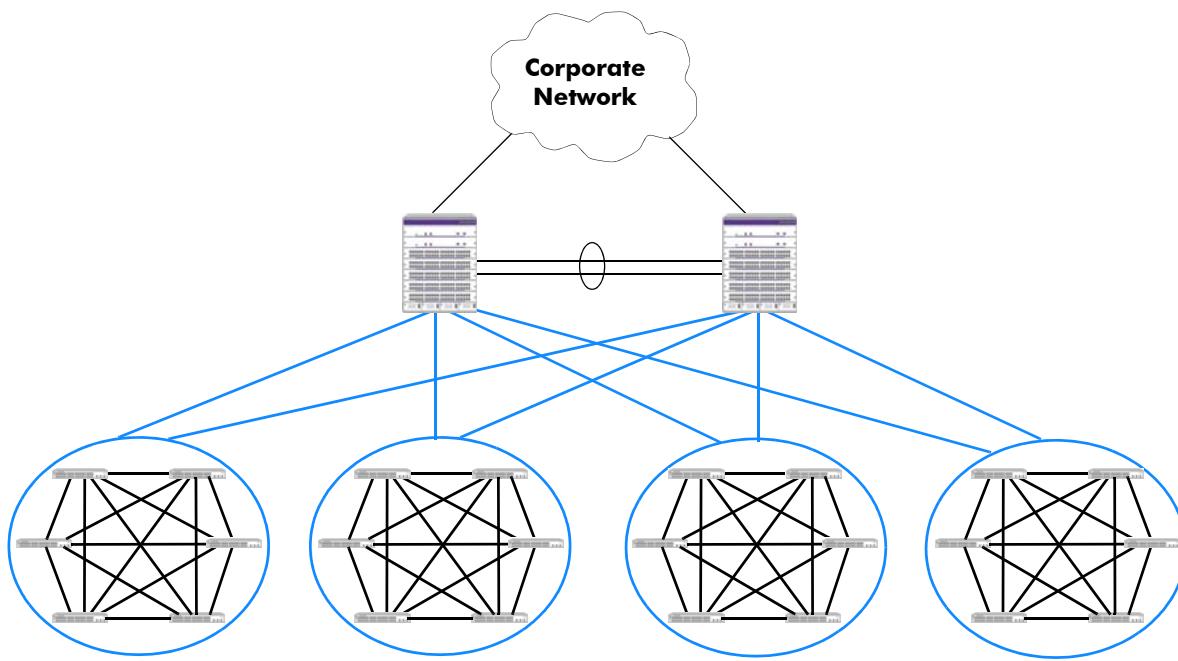
Figure 2: Data Center Mesh

Virtual Network Profile

The Virtual Network Profile (vNP) feature facilitates the discovery and movement of virtual machines (VMs). A vNP is a type of Universal Network Profile (UNP) that is configured specifically for machine classification, especially VMs. A virtual machine is bound to a vNP to ensure consistent application of network access controls and QoS policies when a virtual machine is initially detected or moved to a different network location.

A vNP classifies VMs in the same manner as any other device connected to a UNP port. Once a VM is assigned to a vNP, the VM traffic is bound to the VLAN or service defined in the profile. In addition, any optional QoS policies associated with the profile are also applied to the VM traffic.

Network Virtualization Technology

OmniSwitch data center switching leverages the use of the following AOS Release 8 software features to facilitate a network virtualization solution:

- **Data Center Bridging** (**DCB**) protocols to convert Ethernet into a lossless transport to support a reliable storage area network fabric within the data center mesh. DCB protocols are implemented using embedded profiles in the same manner that QoS QSet profiles are applied.

  The DCB profiles are based on the 802.1Q-REV/D1-5 standard to define how the switch classifies different traffic types and priority mappings and then groups those types into traffic classes. Profiles also specify the Traffic Class Flow (TCF), which is LL (lossless; PFC initiated upstream) or nL (lossy; PFC not initiated upstream).

  For more information about DCB, see Chapter 2, "Configuring Data Center Bridging."

- **Fibre Channel over Ethernet (FCoE) network convergence** solutions to facilitate the expansion of a Fibre Chanel (FC) storage area network (SAN) across an existing Ethernet infrastructure, without having to purchase or manage additional, costly FC equipment. FCoE convergence features supported include the following:

  – **FCoE transit switch.** The OmniSwitch supports the FCoE technology used to tunnel FC frames encapsulated within Ethernet MAC frames. To provide the necessary FCoE transit switch functionality, the OmniSwitch supports FCoE Initialization Protocol (FIP) snooping and Data Center Bridging (DCB) protocols for lossless Ethernet. A transit switch is basically a Layer 2 DCB switch that bridges encapsulated FCoE traffic over the Ethernet fabric between FCoE end devices.

  – **FCoE/FC gateway switch**. The OmniSwitch serves as an FCoE forwarder to connect FCoE nodes to FC switches, connect FC nodes to an FCoE forwarder, and connect native FC fabrics across an FCoE network. Providing this type of functionality allows the OmniSwitch to transparently connect FCoE and FC nodes with an FC SAN across an FCoE (lossless Ethernet) network.

  For more information about the OmniSwitch implementation of FCoE solutions, see Chapter 6, "Configuring FIP Snooping," and Chapter 7, "Configuring an FCoE/FC Gateway."

- **Shortest Path Bridging MAC (SPBM)** to virtualize the data center mesh.

  SPBM extends Layer 2 across the data center mesh while maintaining a loop-free network. All connections between all switches in the topology remain active (no blocking of redundant links).

  SPBM uses the Provider Backbone Bridge (PBB) network model to encapsulate (using IEEE 802.1ah headers) and tunnel customer traffic through the network backbone. The shortest path trees upon which the PBB network infrastructure operates are determined using a version of the Intermediate System-to-Intermediate System (IS-IS) link state protocol that supports TLV extensions for SPB (ISIS-SPB).

  For more information about SPBM, see Chapter 3, "Configuring Shortest Path Bridging."

- **Virtual Chassis (VC)** is a group of chassis managed through a single management IP address.

  VC provides both node level and link level redundancy for devices connecting to the aggregation layer via standard 802.3ad link aggregation mechanisms. See the "Configuring Virtual Chassis" chapter in the *OmniSwitch AOS Release 8 Switch Management Guide* for more information.

- **Virtual eXtensible Local Area Network (VxLAN)** is a Layer 2 overlay network that is used to segment and tunnel device traffic through a data center or cloud network infrastructure. The VXLAN feature is similar to other tunneling and network virtualization solutions in that an encapsulation technique is used to tunnel device traffic through the network. The technique implemented with the VXLAN feature encapsulates an Ethernet MAC frame into an IP packet with a UDP header, then forwards the packet on a Layer 3 network.

  Configuring VXLAN components on the OmniSwitch allows the switch to operate as a VXLAN gateway device. This type of device connects VXLAN and non-VXLAN (traditional VLAN) segments. See Chapter 4, "Configuring a VXLAN Gateway," for more information.

- **VXLAN Snooping** attempts to detect and identify VXLAN traffic by sampling packets to determine if they are VXLAN encapsulated packets. Once this type of traffic is identified, VXLAN Snooping collects and stores information about the flow in a database on the local switch. Additional configurable options for this feature include the ability to apply QoS policy list rules to the identified flow and SNMP trap generation. See Chapter 5, "Configuring VXLAN Snooping," for more information.

Data Center Mesh Configuration Example

Traditional data center networks consist of a 3-tier approach: access, aggregation, and core. The high-speed, low-latency capability of the OmniSwitch provides the ability to combine the access and aggregation tiers.

The application example in this section consists of two data centers that incorporate the use of the OmniSwitch data center switching mesh architecture. Each pod shown in Figure 5 interconnects six switches delivering 240 server-facing ports while maintaining low latency between servers in the same pod.

In this example, each OmniSwitch pod is connected to two OmniSwitch core switches to form the data center mesh. In turn, the OmniSwitch core switches connect to Service Routers (7750) platforms to provide interconnectivity between the two data centers through an MPLS cloud over Layer 2 virtual LAN services.

The following diagram illustrates the example data center switching network that implements the OmniSwitch pod/mesh architecture combined with supported virtualization technologies.
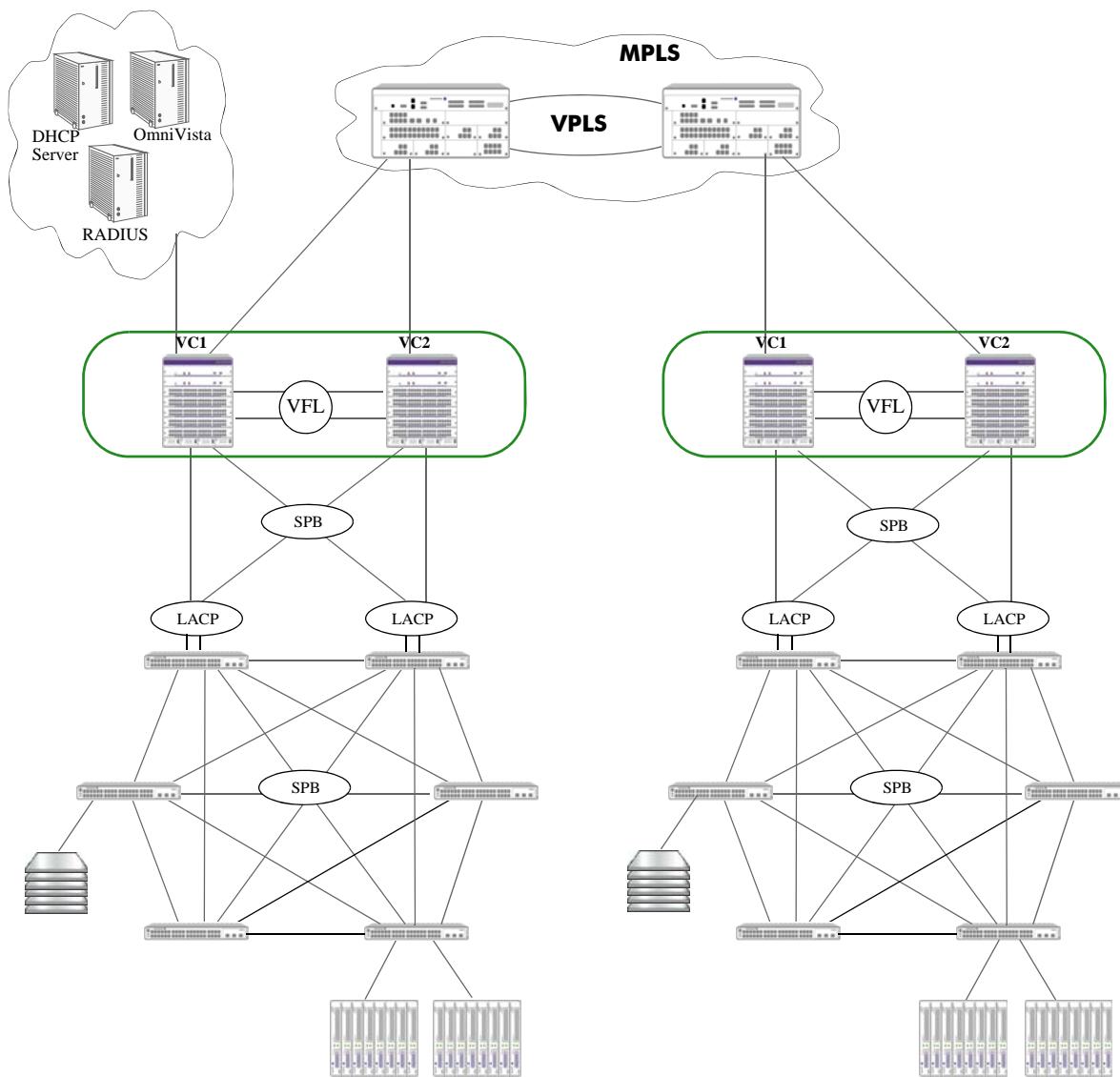
Figure 5: Data Center Mesh Example

In this data center mesh configuration example:

● Two separate data center configurations are depicted, each using a single OmniSwitch pod and two OmniSwitch core switches.

● Each pod consists of six switches. The pod is formed by connecting each switch to every one of the other six switches. In other words, each switch is configured with five links that connect the switch to the pod.

● The core tier for each data center is made up of two OmniSwitch chassis-based switches. Virtual Chassis (VC) is configured for each pair of chassis-based switches to create a chassis group that is managed through a single management IP address. VC provides both node level and link level redundancy for the chassis group connecting to the aggregation (pod) layer via standard 802.3ad link aggregation mechanisms.

The following configuration snapshot shows a sample VC configuration for the core switches:

```
VC1-> show configuration vcm-snapshot chassis-id 1
! Virtual Chassis Manager:
virtual-chassis chassis-id 1 configured-chassis-id 1
virtual-chassis chassis-id 1 vf-link 0 create
virtual-chassis chassis-id 1 vf-link 0 member-port 1/2/1
```

- Shortest Path Bridging MAC (SPBM) is used in each data center mesh (pod plus core) to define sets of loop-free shortest path trees (SPTs) through the network. Each switch serves as the SPT root for all traffic entering the switch, thus allowing the switch to provide the shortest path to every other switch.

  The bridging methodology that allows each switch to serve as its own root is enforced through the use of SPBM backbone VLANs (BVLANs). The BVLAN is a transport VLAN for SPBM services and SPT calculations.

  The following configuration snapshot shows a sample SPBM bridging configuration. Each switch that will participate in the SPBM domain (pod and core) must have the same BVLAN configuration. SPBM merges the core and pod together into the same virtual mesh domain.

```
-> show configuration snapshot spb-isis
! SPB-ISIS:
spb isis bvlan 4001 ect-id 1
spb isis bvlan 4002 ect-id 2
spb isis control-bvlan 4001
spb isis interface port 1/1/2-5
spb isis interface port 1/1/7
spb isis interface port 2/1/2-5
spb isis interface linkagg 2
spb isis admin-state enable
```

- SPBM Ethernet services are bound to the SPT bridging configuration and are used to encapsulate and tunnel data through the data center mesh. SPBM services are configured on any OmniSwitch in the mesh that will originate or terminate a service. This usually is on switches that are connected to host devices or that connect to other networks where traffic will enter or leave the mesh.

  The OmniSwitch Service Manager feature is used to build the SPBM service architecture layer within the SPBM mesh domain. Provisioning a service requires the configuration of three logical entities: a service instance identifier (I-SID), service access port, and a service access point (SAP).

  The following configuration snapshot shows a sample SPB service configuration:

```
-> show configuration snapshot svcmgr
! SVCMGR:
service access port 1/11
service access port 1/12
service access port 1/13
service spb 1001 isid 1001 bvlan 4001 admin-state enable
service spb 1002 isid 1002 bvlan 4001 admin-state enable
service spb 1001 sap port 1/11:1001 admin-state enable
service spb 1001 sap port 1/12:1001 admin-state enable
service spb 1001 sap port 1/13:1001 admin-state enable
service spb 1002 sap port 1/11:1002 admin-state enable
service spb 1002 sap port 1/12:1002 admin-state enable
service spb 1002 sap port 1/13:1002 admin-state enable
```

- When a physical switch port comes up, a QSet instance (a set of eight queues) is automatically associated with the port for unicast traffic. In addition, a default Data Center Bridging profile (DCP 8) is automatically assigned to the QSI.

The DCP 8 profile applies strict priority scheduling with lossy traffic class management to each of the eight egress port queues. However, ports connected to devices running critical applications that require lossless Ethernet, such as the storage and server hosts shown in Figure 5, can be assigned to one of the other 10 pre-defined profiles or a user-configured profile that will provide the necessary class of service for that device.

The following configuration snapshot shows a sample DCB configuration. In this sample, a custom profile (DCP 11) is created with custom settings for the traffic classes. DCP 11 is then assigned to ports 1/1 and 1/9-10, replacing the default DCP 8 assignment. In addition, ports 1/12 and 1/34-35 are assigned to DCP 7 and 9. All other ports on the switch are using default DCP 8.

```
-> show configuration snapshot vfc
! Virtual Flow Control:
qos qsp dcb 11 import qsp dcb "dcp-9"
qos qsp dcb "dcp-11" tc 1 pfc flow-type nLL
qos qsp dcb "dcp-11" tc 0 pfc flow-type nLL
qos qsp dcb "dcp-11" tc 2 pfc flow-type nLL
qos qsp dcb "dcp-11" tc 4 pfc flow-type nLL
qos qsp dcb "dcp-11" tc 5 pfc flow-type nLL
qos qsp dcb "dcp-11" tc 6 pfc flow-type nLL
qos qsp dcb "dcp-11" tc 7 pfc flow-type nLL
qos qsi port 1/1 qsp dcb "dcp-11"
qos qsi port 1/9-10 qsp dcb "dcp-11"
qos qsi port 1/12 qsp dcb "dcp-7"
qos qsi port 1/34-35 qsp dcb "dcp-9"
```

- Virtual Network Profile (vNPs) are configured on each switch that is connected to a server to facilitate VM discovery and mobility within the local data center or mobility between the local and remote data center. In addition, Universal Network Profile (UNP) functionality is enabled on each of the server connections to activate vNP classification of VMs.

  The vNPs are used to assign VMs to SPBM services that span the data center mesh. When a VM is discovered or migrates to a new location, the assigned vNP applies the necessary access controls and any QoS policies specifically defined for the VM.

  The following configuration snapshot shows a sample vNP configuration.

```
-> show configuration snapshot da-unp
! DA-UNP:
unp domain 1 description "AT&T"
unp domain 2 description "Sprint"
unp port 1/15 port-type access
unp port 1/15 domain 1
unp port 1/16 port-type access
unp port 1/16 domain 1
unp port 1/17 port-type access
unp port 1/17 domain 2
unp port 1/18 port-type access
unp port 1/18 domain 2
unp profile unp_for_spb1
unp profile unp_for_spb1 map service-type spb tag-value 115:150 isid 6000 bvlan
4001
unp classification vlan-tag 125 profile1 unp_for_spb1
unp classification vlan-tag 225 profile1 unp_for_spb1
```

2 Configuring Data Center Bridging

Data Center Bridging (DCB) is a set of standards that extend Ethernet capabilities to support the convergence of storage and data in today's virtualized networks. The OmniSwitch implementation of DCB supports the following DCB standards:

- **Priority-Based Flow Control (PFC)**—based on the IEEE 802.1Qbb standard, PFC pauses traffic based on congestion priority instead of blocking the entire link when congestion occurs. Allows lossless and lossy traffic with different priorities on the same physical port.

- **Enhanced Transmission Selection (ETS)**—based on the IEEE 802.1Qaz standard, ETS groups related traffic into priority groups (traffic classes) to which bandwidth guarantees and scheduling are applied.

- **Data Center Bridging Exchange (DCBX)**—exchanges and negotiates PFC and ETS information between two directly connected peer devices.

This implementation of the DCB enhanced Ethernet protocols uses embedded profiles to apply the PFC, ETS, and DCBX configuration to traffic flows. This approach is similar to how QSet profiles (QSPs) are used to apply the QoS configuration for bandwidth management and egress port queue scheduling.

In This Chapter

This chapter describes DCB in general and how DCB profiles and port configurations are applied to the switch. It provides information about configuring DCB through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following topics and procedures are included in this chapter:

DCB Defaults

Traffic class management and related QoS functions are implemented using a Queue Set (QSet) framework. Each port and link aggregate is associated with a set of eight egress queues, referred to as a Queue Set Instance (QSI). Each QSI is associated with DCB profile 8 (DCP 8) by default.

A DCP defines both the Data Center Bridging Exchange (DCBX) control parameters and the traffic class management parameters that are applied to the eight queues associated with the QSI. See "Configuring DCB Profiles" on page 2-20 for more information.

The following are the default DCBX and traffic class management settings applied with DCP 8:

<table>
<tr><td colspan="8" align="center">DCB Profile 8 (Default Profile)</td></tr>
<tr><td colspan="8" align="center">Profile Control :: DCBX Configuration</td></tr>
<tr><td colspan="4">PFC :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Capability : 8

ETS :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Max TC : 3</td><td colspan="4">Recommendation :: Enable TLV : [*Yes/No]
Recommended Profile : [DCB_Def_Profile_1/NULL]

Application :: Enable TLV : [Yes/*No]
Defense Mode :: [*Enable/Disable]
Default MTU :: [9216]</td></tr>
<tr><td colspan="8" align="center">Profile Data :: Max Traffic Classes (TC) : 8</td></tr>
<tr><td>Traffic Class (TC)</td><td>Traffic Type (TT)</td><td>TC Description (TD)</td><td>TC Priority Map (TP)</td><td colspan="2">TC Bandwidth (%)
Min   Max</td><td>TC Flow Control (TF)</td><td>TC Scheduler (TS)</td></tr>
<tr><td>0</td><td>BE</td><td>Best Effort</td><td>0</td><td>0</td><td>100</td><td>Lossy</td><td>Strict Priority</td></tr>
<tr><td>1</td><td>BK</td><td>Background</td><td>1</td><td>0</td><td>100</td><td>Lossy</td><td>Strict Priority</td></tr>
<tr><td>2</td><td>EE</td><td>Excellent Effort</td><td>2</td><td>0</td><td>100</td><td>Lossy</td><td>Strict Priority</td></tr>
<tr><td>3</td><td>CA</td><td>Critical Applications</td><td>3</td><td>0</td><td>100</td><td>Lossy</td><td>Strict Priority</td></tr>
<tr><td>4</td><td>VI</td><td>Video (<100ms latency)</td><td>4</td><td>0</td><td>100</td><td>Lossy</td><td>Strict Priority</td></tr>
<tr><td>5</td><td>VO</td><td>Voice (<10ms latency)</td><td>5</td><td>0</td><td>100</td><td>Lossy</td><td>Strict Priority</td></tr>
<tr><td>6</td><td>IC</td><td>Internetwork Control</td><td>6</td><td>0</td><td>100</td><td>Lossy</td><td>Strict Priority</td></tr>
<tr><td>7</td><td>NC</td><td>Network Control</td><td>7</td><td>0</td><td>100</td><td>Lossy</td><td>Strict Priority</td></tr>
</table>

Note. QSet profiles and DCB profiles are mutually exclusive on the same port. If the OmniSwitch Data Center software license is installed, then DCB profiles are used by default. If this license is not installed, then QSet profiles are used by default. See "Using DCB Profiles" on page 2-14 for more information.

DCB Port Defaults

Parameter Description	Command	Default
The DCBX administrative status for the port.	**qos qsi dcb dcbx admin-state**	Enabled
The transmission status for the PFC configuration TLV	**qos qsi dcb dcbx pfc tlv**	Enabled
The PFC defense mode status	**qos qsi dcb dcbx pfc defense**	Enabled
The PFC "willing" to negotiate status	**qos qsi dcb dcbx pfc willing**	Yes
The transmission status for the ETS configuration TLV	**qos qsi dcb dcbx ets config-tlv**	Enabled
The transmission status for the ETS recommended TLV	**qos qsi dcb dcbx ets recommended TLV**	Enabled
The ETS "willing" to negotiate status	**qos qsi dcb dcbx ets willing**	Yes

Data Center Bridging Overview

Convergence of data and storage into Ethernet has become more prevalent due to the combined data and storage bandwidth requirements of virtualized networks. Data Center Bridging (DCB) technology combined with the Alcatel-Lucent Enterprise Data Center Mesh architecture provides a unified framework upon which different types of network traffic are converged into a single transport layer.

For example, each of the following traffic types can coexist in a converged network without imposing serious restrictions on the performance of the other traffic types:

* Storage Area Network (SAN) traffic—A SAN provides access to a shared storage solutions. Requires transmission of traffic with no packet loss (Lossless).

* Local Area Network (LAN) traffic—A LAN provides end-user access to server-based network applications, such as email, database, and web servers. Requires best effort transmission of traffic for important applications. Can tolerate some traffic loss (Lossy).

* Inter-Process Communication (IPC) traffic—IPC refers to the mechanisms or activities that provide communications and data sharing between applications. Requires the highest priority for low latency transmission.

The OmniSwitch supports the following Data Center Bridging (DCB) protocols to provide a single, shared infrastructure for reliable delivery of data and storage over Ethernet:

* **Priority-Based Flow Control (PFC)**—Based on the IEEE 802.1Qbb standard, PFC pauses traffic based on congestion priority instead of blocking the entire link when congestion occurs. Allows lossless and lossy traffic with different priorities on the same physical port.

* **Enhanced Transmission Selection (ETS)**—Based on the IEEE 802.1Qaz standard, ETS provides a common framework for dynamic bandwidth management. ETS groups related traffic into priority groups (Traffic Classes) to which bandwidth guarantees and scheduling are applied.

* **Data Center Bridging Exchange (DCBX)**—Based on the IEEE 802.1Qaz standard, DCBX uses the Link Layer Discovery Protocol (LLDP) to exchange and negotiate PFC and ETS information between two directly connected peer switches. Enabled by default, DCBX is responsible for auto-negotiation and auto-configuration of link parameters for DCB functions.

Priority-Based Flow Control Overview

Priority-based Flow Control (PFC) is the evolution of the IEEE 802.3 Ethernet PAUSE mechanism that uses pause frames to signal the other end of a link to stop sending traffic when buffer congestion is detected on any one of the ingress port queues. The problem with the 802.3x approach is that when congestion occurs for only one type of traffic on a specific queue, the transmission of all traffic on the port is stopped.

PFC offers a more granular approach that is needed to support the convergence of various traffic types over the same link. Instead of pausing all traffic when ingress port congestion occurs, PFC only pauses traffic tagged with a specific Class of Service (CoS) priority value while allowing traffic assigned to other priority values to continue (or get dropped when congestion occurs) on the same link.

In essence, using PFC divides an Ethernet link into eight virtual lanes. Each virtual lane represents one of the CoS priority values (0–7). When buffer congestion occurs for traffic marked with one of these priority values, PFC initiates pause frames back to the source of that specific traffic. The source then holds back the traffic until the congestion clears (bandwidth allocated for that priority becomes available). As shown in Figure 1, PFC detected congestion for traffic marked with priority 4, so only that traffic is paused and

packet loss is avoided. All other traffic marked with different priorities continues to forward on the link until congestion occurs for those priorities as well.

Ethernet Link
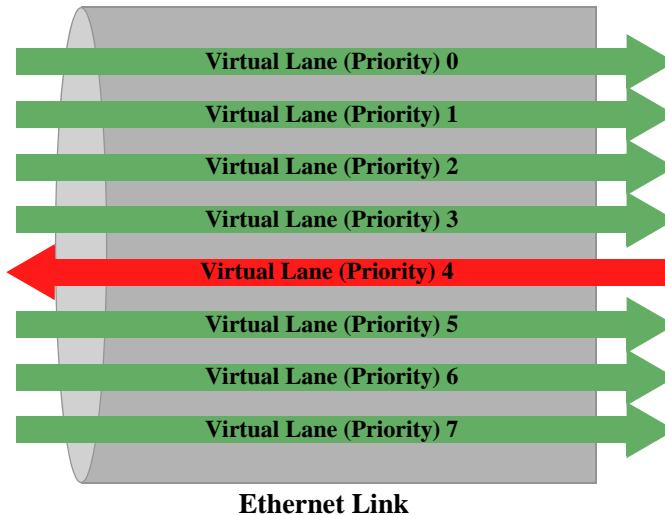
Figure 1: PFC Example

To determine how traffic is treated when assigned to one of these eight priority values, the pre-defined OmniSwitch DCB profiles use the following general IEEE recommendations for mapping traffic types to the CoS priority values:

- Priority 1, BK—Background {Bulk transfers}

- Priority 0, BE—Best Effort {unprioritized user applications}

- Priority 2, EE—Excellent Effort {CEO's best effort}

- Priority 3, CA—Critical Applications {guaranteed minimum bandwidth}

- Priority 4, VI—Video {low latency < 100ms}

- Priority 5, VO—Voice {low latency < 10ms}

- Priority 6, IC—Internetwork Control {guaranteed delivery}

- Priority 7, NC—Network Control {guaranteed delivery}

Using PFC for OmniSwitch Flow Control

The main purpose of PFC is to provide a lossless transport for traffic sensitive to packet loss, such as storage area network (SAN), LAN backup, or management traffic, while at the same time allowing packet-drop congestion management for other traffic types on the same link. The OmniSwitch implementation of DCB accomplishes this through the use of embedded profiles that define the PFC configuration for each priority queue associated with each switch port.

There are 11 pre-defined profiles available, and the ability to create additional custom profiles is also supported. Each profile specifies which priority queue requires lossless (no packet drop) or lossy (packet drop) congestion management. When a switch port comes up, a single, pre-defined DCB profile is associated with that port by default. Currently DCB profile 8 is the default profile applied to each port. This profile defines lossy flow control for all eight queues (virtual lanes).

The following general steps are required to configure a lossless transport lane for a specific type of traffic:

1 Select a pre-defined DCB profile or create a custom profile that defines lossless flow control for a specific CoS priority value.

2 Determine the end-to-end lossless data path through the OmniSwitch network and apply the profile identified in Step 1 to each port in that path.

3 Make sure that the traffic requiring lossless transmission is marked with the same CoS priority value to which the DCB profile will apply lossless flow control. The switch will only pause frames sent with the same priority value as the lossless priority value specified in the DCB profile.

4 Make sure that all ingress ports are configured as QoS trusted ports with the default classification set to 802.1p.

For more information about DCB profiles, see "Using DCB Profiles" on page 2-14.

For more information about CoS 802.1p priority bit classification and marking, see "How Traffic is Classified and Marked" on page 26-5 in Chapter 26, "Configuring QoS," of the *OmniSwitch AOS Release 8 Network Configuration Guide*.

Enhanced Transmission Selection Overview

Priority-based Flow Control (PFC) uses the Class of Service (COS) 802.1p priority values (0–7) to determine which traffic to pause when congestion occurs. Enhanced Transmission Selection (ETS) groups these priorities (traffic types) into Traffic Classes and then allocates a percentage of minimum bandwidth to each class.

Traffic Classes are groups of priorities that have similar traffic handling requirements (for example, LAN, SAN, and IPC). The amount of bandwidth assigned to each Traffic Class is a minimum guarantee of available bandwidth. Available bandwidth is defined as the amount of link bandwidth remaining after Traffic Classes not subject to ETS (for example, Strict Priority is used instead) or vendor-specific requirements are serviced. In addition,

* Available bandwidth for a Traffic Class is configurable in 1% increments.

* The maximum deviation of available bandwidth is 10% when all of the following is true:
  – Only ETS traffic is flowing.
  – No PFC pause frames have been received.
  – All ETS Traffic Classes are offered enough load to consume their share of allocated bandwidth.

* Any unused portion of the minimum bandwidth is made available for use by other classes, but only until the original Traffic Class needs the bandwidth again. At such time, PFC (if enabled) will pause the required traffic to minimize packet loss until the bandwidth is regained.

Using ETS for OmniSwitch Bandwidth Allocation

The OmniSwitch implementation of DCB uses embedded profiles to define the ETS and PFC configuration for each priority queue associated with each switch port. ETS combines the CoS priorities used by PFC into Traffic Classes. Each DCB profile defines the ETS grouping of such priority values into Traffic Classes to which the specified bandwidth allocation, flow control (lossless or lossy), and scheduling are applied.

ETS requires a minimum of three Traffic Classes per port but supports a maximum of eight Traffic Classes per port. The use of DCB profiles to apply the ETS configuration facilitates the correct mapping of traffic types and priorities into Traffic Classes according to the IEEE 802.1Q-REV/D1-5 standard.

There are 11 pre-defined DCB profiles available, and the ability to create additional custom profiles is also supported. The following table shows the grouping of traffic types into a Traffic Class (TC) as applied by the pre-defined DCB profiles (see "Using DCB Profiles" on page 2-14).

Table 1: DCB Profile Traffic Classes

DCB Profile	Traffic Type Allocation	TC Priority Map
1	**TC-0**: Best Effort, Background, Excellent Effort, Critical Applications **TC-1**: Voice, Video **TC-2**: Network Control, Internetwork Control	0, 1, 2, 3 5, 4 7, 6
2	**TC-0**: Best Effort, Background **TC-1**: Critical Applications, Excellent Effort **TC-2**: Voice, Video **TC-3**: Network Control, Internetwork Control	0, 1, 3, 2 5, 4 7, 6
3	**TC-0**: Best Effort, Background **TC-1**: Critical Applications, Excellent Effort **TC-2**: Voice, Video **TC-3**: Internetwork Control **TC-4**: Network Control	0, 1, 3, 2 5, 4 6 7
4	**TC-0**: Background **TC-1**: Best Effort **TC-2**: Critical Applications, Excellent Effort **TC-3**: Voice, Video **TC-4**: Internetwork Control **TC-5**: Network Control	1 0 3, 2 5, 4 6 7
5	**TC-0**: Background **TC-1**: Best Effort **TC-2**: Excellent Effort **TC-3**: Critical Applications **TC-4**: Voice, Video **TC-5**: Internetwork Control **TC-6**: Network Control	1 0 2 3 5, 4 6 7
6	**TC-0**: Background **TC-1**: Best Effort **TC-2**: Excellent Effort **TC-3**: Critical Applications **TC-4**: Video **TC-5**: Voice **TC-6**: Internetwork Control **TC-7**: Network Control	1 0 2 3 4 5 6 7

DCB Profile	Traffic Type Allocation	TC Priority Map
7	**TC-0**: Best Effort	0
	TC-1: Background	1
	TC-2: Excellent Effort	2
	TC-3: Critical Applications	3
	TC-4: Video	4
	TC-5: Voice	5
	TC-6: Internetwork Control	6
	TC-7: Network Control	7
8	**TC-0**: Best Effort	0
	TC-1: Background	1
	TC-2: Excellent Effort	2
	TC-3: Critical Applications	3
	TC-4: Video	4
	TC-5: Voice	5
	TC-6: Internetwork Control	6
	TC-7: Network Control	7
9	**TC-0**: Background	1
	TC-1: Best Effort	0
	TC-2: Excellent Effort	2
	TC-3: Critical Applications	3
	TC-4: Video	4
	TC-5: Voice	5
	TC-6: Internetwork Control	6
	TC-7: Network Control	7
10	**TC-0**: Background	1
	TC-1: Best Effort	0
	TC-2: Excellent Effort	2
	TC-3: Critical Applications	3
	TC-4: Video	4
	TC-5: Voice	5
	TC-6: Internetwork Control	6
	TC-7: Network Control	7

Figure 2 further illustrates ETS Traffic Class groupings by showing the Traffic Class minimum bandwidth guarantees as defined and applied by DCB profile 2:
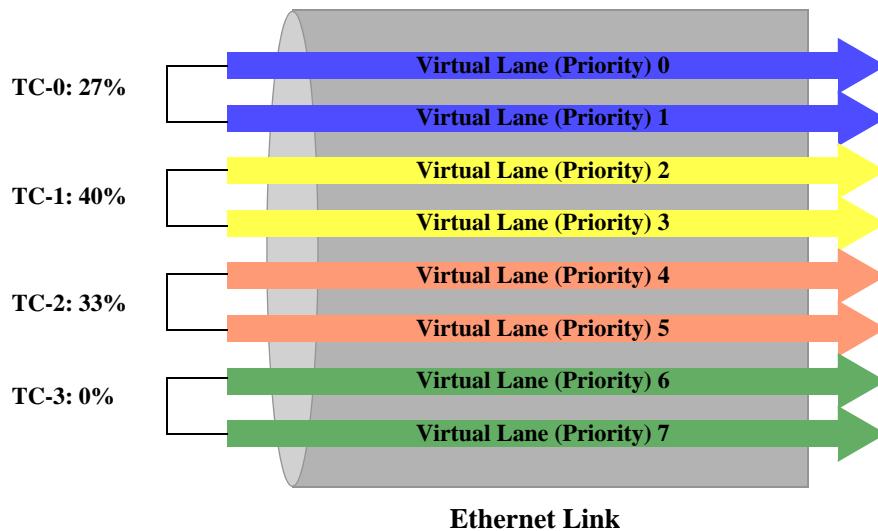
Figure 2: ETS Traffic Classes (DCB Profile 2)

In the Figure 2 example,

- Traffic marked with priorities 0 and 1 is grouped into Traffic Class 0, which is assigned a minimum bandwidth guarantee of 27% of the available bandwidth.

- Traffic marked with priorities 2 and 3 is grouped into TC-1, which is assigned a minimum bandwidth guarantee of 40% of the available bandwidth.

- Traffic marked with priorities 4 and 5 is grouped into TC-2, which is assigned a minimum bandwidth guarantee of 33% of the available bandwidth.

- Traffic marked with priorities 6 and 7 is grouped into TC-3, but the TC-3 minimum bandwidth is set to zero. This is because DCB profile 2 applies Strict Priority scheduling to TC-3, not ETS. As a result, virtual lanes (priority queues) 6 and 7 are serviced first then ETS allocates the remaining bandwidth for TC-0, TC-1, and TC-2 priority queues.

For more information about DCB profiles, see "Using DCB Profiles" on page 2-14.

For more information about 802.1p CoS priority bit classification and marking, see "How Traffic is Classified and Marked" on page 26-5 in Chapter 26, "Configuring QoS," of the *OmniSwitch AOS Release 8 Network Configuration Guide*.

Data Center Bridging Exchange Overview

The Data Center Bridging Capabilities Exchange Protocol (DCBX) communicates the DCB capabilities of each OmniSwitch and exchanges PFC and ETS configuration details with directly connected peer devices. In addition, DCBX functionality helps to ensure a consistent configuration across the network.

- DCBX discovers the DCB capabilities of directly connected peers (for example, does the port on the other end of the link support PFC and is it active).

- Mis-configurations are detected when DCB configuration parameter values are exchanged between two peers and the parameters are not the same on both peers.

- When a configuration mismatch is detected and both peers are "willing", an automatic negotiation takes place to reconcile a common operational configuration for both peers.

DCBX uses the IEEE 802.1AB Link Layer Discover Protocol (LLDP) to discover and exchange information between two link peers. There are specific LLDP Type-Length-Values (TLVs) that DCBX defines for exchanging PFC and ETS attribute values, such as:

- Is PFC in use on the peer switch.

- What are the priority values that PFC manages.

- Which traffic type priority values are mapped to a specific Traffic Class.

- What are the minimum bandwidth guarantees defined for each Traffic Class.

An OmniSwitch is considered a DCB-capable switch if the OmniSwitch Data Center software license is installed. When this license is installed, DCBX is automatically enabled on each switch port and DCB profiles apply the PFC and ETS configuration to the QSet instance (QSI) for each port.

The following default DCBX configuration is applied to each DCBX-enabled port:

- PFC and ETS TLVs are enabled. The following TLVs are supported:
  - PFC Configuration
  - ETS Configuration
  - ETS Recommendation

- The "willing" bit setting is enabled. This setting indicates that the switch is willing to negotiate changes to the operational DCBX configuration on the local port to match the DCBX configuration on the remote port.

- PFC defense mode is enabled. When PFC negotiations fail and the defense mode is enabled, no PFC functionality is applied to any priority but traffic flow is allowed to continue. However, if the defense mode is disabled when negotiations fail, the local PFC configuration is applied.

DCBX in the Data Center Mesh

The diagram in Figure 3 shows where DCBX is enabled in a sample OmniSwitch Data Center Mesh topology. In this example,

- The OmniSwitch Data Center Software License is installed on each OmniSwitch to enable the DCB features (PFC, ETS, and DCBX) on each switch port.

- DCBX negotiates PFC and ETS parameters between the switches on every port. Manual configuration between an OmniSwitch and a non-OmniSwitch device (servers or switches from other vendors) may require manual configuration of DCB parameters on those devices.

• The servers could be sending SAN traffic at a priority negotiated for lossless and could use a different priority for iSCSI SAN traffic. Note that any application requiring lossless access to the network is supported as long as the traffic is stamped with the correct priority that is negotiated for lossless.
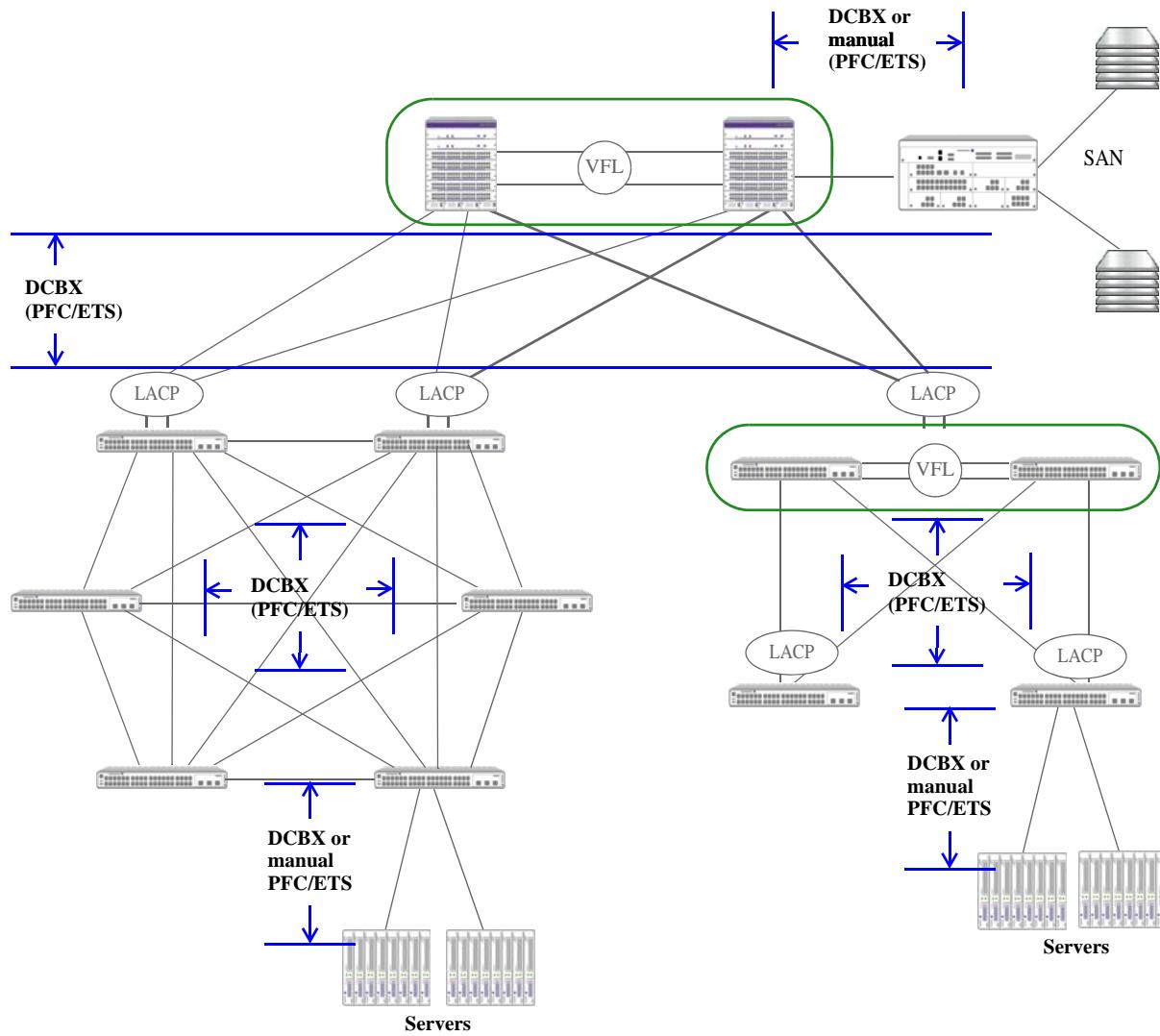
Figure 3: DCBX Example

For more information about DCBX, see "Configuring DCBX Port Parameters" on page 2-23.

For more information about DCB profiles, see "Using DCB Profiles" on page 2-14.

Interaction with Other Features

This section contains important information about how other OmniSwitch features interact with Data Center Bridging (DCB) features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

Converged Enhanced Ethernet DCBX

The OmniSwitch implementation of Data Center Bridging supports two versions of the DCB Exchange protocol (DCBX):

- IEEE 802.1Qaz DCBX

- Converged Enhanced Ethernet (CEE) DCBX version 1.0.1

By default, a DCB port will use the IEEE 802.1Qaz version of DCBX until the port detects the peer switch is using the CEE version. When this occurs, the switch will automatically stop 802.1Qaz DCBX on the port and start using CEE DCBX.

It is possible to configure the DCB port to use only a specific version of DCBX (no auto-detection) using the **qos qsi dcb dcbx version** command. For example, the following command changes the DCBX version for the specified port to CEE:

```
-> qos qsi port 1/10 dcb dcbx version cee
```

To change the DCBX version to IEEE 802.1Qaz, use the **qos qsi dcb dcbx version** command with the **ieee** parameter. For example:

```
-> qos qsi port 1/11 dcb dcbx version ieee
```

To change the port back to automatically detecting the peer version, use the **qos qsi dcb dcbx version** command with the **auto** parameter. For example:

```
-> qos qsi port 1/11 dcb dcbx version auto
```

Note. Using DCB profile 11 (DCP 11) is highly recommended when connecting an OmniSwitch port to equipment from other vendors. For more information see "Using DCB Profiles" on page 2-14 and "Configuring DCB Profiles" on page 2-20.

QoS

- This implementation of DCB provides enhanced QoS congestion and bandwidth allocation to support multiple traffic types on the same Ethernet link. However, DCB profiles are used to apply the DCB configuration instead of QoS profiles. These two profile types are mutually exclusive on the same port; only one or the other profile type is applied to a port at any given time.

- Even though a port is only associated with a single DCB profile or a single QoS profile, it is possible to have a mixture of both profile types on different ports within the same switch. For example, when a DCB-licensed switch boots up, all ports are associated with a DCB profile by default, but the profile association for any port can be changed to a QoS profile without disrupting the DCB profile associations of other ports.

- The DCB protocols, Priority-based Flow Control (PFC) and Enhanced Transmission Selection (ETS), use the Class of Service (CoS) 802.1p markings found in the frame header to define traffic groups.

These markings are the result of QoS classification that occurs prior to and separately from the application of DCB functionality. DCB does not replace existing QoS classification and enqueuing mechanisms.

For more information about DCB and QoS profiles, see "Using DCB Profiles" on page 2-14 and "Multicast and Unicast Traffic Distribution" on page 2-24.

For more information about 802.1p CoS priority bit classification and marking, see "How Traffic is Classified and Marked" on page 26-5 in Chapter 26, "Configuring QoS," of the *OmniSwitch AOS Release 8 Network Configuration Guide*.

Shortest Path Bridging

In order to support DCBX on an SPB access port (for example, between a server and an OmniSwitch), create a Layer 2 profile to allow the access port to participate as an LLDP (802.1ab) protocol peer to the server. If this is not done, then LLDP packets will be treated as traffic and DCBX will not be negotiated between the access port and server.

The following is an example of an SPB configuration that creates Layer 2 profile "DCBX" and associates that profile with access port 1/1/6:

```
-> show configuration snapshot svcmgr
! SVCMGR:
service l2profile DCBX 802.1ab peer
service access port 1/1/6 l2profile DCBX
```

SNMP

If SNMP performs sets on the standard MIBs at the port level and the port is associated with:

- A custom DCB profile that is only associated with that one port, then the changes are made on that profile.

- A custom DCB profile that is associated with multiple ports, a new custom profile is created with the changes and the new profile is associated only with that port. The new profile is accessible through the OmniSwitch Command Line Interface (CLI).

- A default DCB profile, a new custom profile is created with the changes and the new profile is associated only with that port. The new profile is accessible through the OmniSwitch CLI.

For more information about DCB profiles, see "Using DCB Profiles" on page 2-14.

Virtual Chassis

- DCBX is not supported on VF-Links.

- Only two profiles can be applied on the VF-links: DCB 8 (applied by default, all priorities are lossy) and DCB 9 (all priorities are lossless, PFC is enabled).

- When configuring PFC over VFL on an OmniSwitch 6900, the VFL should not have more than eight ports comprising the VFL.

Using DCB Profiles

The DCB configuration for PFC, ETS, and DCBX is applied to switch ports through DCB profiles the same way that QSet profiles apply the QoS queue management configuration to switch ports. However, only DCB profiles or QSet profiles are allowed at any given time.

> **Note.** QSet profiles and DCB profiles are mutually exclusive. If the OmniSwitch Data Center software license is installed, then DCB profiles are used. If this license is not installed, then QSet profiles are used.

- If the OmniSwitch Data Center software license is not installed, the switch boots up as a "non-DCB" switch and QSet profiles are applied to switch ports.

- If the OmniSwitch Data Center software license is installed, the switch boots up as a DCB switch and profiles are applied as follows:
  - If there is no existing DCB configuration, the default DCB profile (DCP 8) is applied to all switch ports. This occurs even if the port was previously assigned to a non-default QSP (for example, QSP 2, 3, or 4).
  - If there is an existing DCB configuration, the profiles are applied based on that configuration.

- If a switch boots up in the DCB mode and no DCB configuration is present (only default DCP 8), the switch will start DCBX by default to make sure each port is auto-configurable via DCBX to match the peer configuration. This provides a "plug-and-play" installation process that allows a switch running the default DCB configuration to automatically adapt to the network.

The DCB profiles are based on the 802.1Q-REV/D1-5 standard to define how the switch classifies different traffic types and priority mappings and then groups those types into traffic classes. Profiles also specify the Traffic Class Flow (TCF), which is LL (lossless; PFC initiated upstream) or nL (lossy; PFC not initiated upstream).

There are 11 pre-defined DCB profiles (DCP 1–11) available that represent common applications of the DCB standards (see "Pre-Defined DCB Profiles (Unicast)" on page 2-14). Creating custom profiles is also allowed; a maximum of 128 (including the 11 pre-defined) profiles per switch is supported.

Pre-Defined DCB Profiles (Unicast)

A DCB profile defines a set of PFC, ETS, and DCBX parameters that are applied to different traffic classes. Using profiles simplifies the DCB configuration process across the network. Once the appropriate pre-defined profile is selected or a custom profile is created for a particular set of traffic classes, the profile is then applied to each OmniSwitch that lies in the transport path for the specified set of traffic classes.

This section contains the Traffic Class priority mappings and the related DCB configuration parameters for each of the eleven pre-defined OmniSwitch DCB profiles. By default, each QSet port instance is associated with DCP 8. See "Multicast and Unicast Traffic Distribution" on page 2-24 for more information.

DCB Profile 1							
Profile Control :: DCBX Configuration							

PFC :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Capability : 8
 	
ETS :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Max TC : 3
			Recommendation :: Enable TLV : [*Yes/No]
			Recommended Profile : [DCB_Def_Profile_1/NULL]
			
			Application :: Enable TLV : [Yes/*No]
			Defense Mode :: [*Enable/Disable]
			Default MTU :: [9216]				

Profile Data :: Max Traffic Classes (TC) : 3							
Traffic Class (TC)	Traffic Type (TT)	TC Description (TD)	TC Priority Map (TP)	TC Bandwidth (%) Min	Max	TC Flow Control (TF)	TC Scheduler (TS)
0	BE	Best Effort	0, 1, 2, 3	33	100	Lossy	ETS
1	VO	Voice (<10ms latency)	5, 4	67	100	Loss Less	ETS
2	NC	Network Control	7, 6	0	100	Lossy	Strict Priority

DCB Profile 2							
Profile Control :: DCBX Configuration							

PFC :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Capability : 8
 	
ETS :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Max TC : 3
			Recommendation :: Enable TLV : [*Yes/No]
			Recommended Profile : [DCB_Def_Profile_1/NULL]
			
			Application :: Enable TLV : [Yes/*No]
			Defense Mode :: [*Enable/Disable]
			Default MTU :: [9216]				

Profile Data :: Max Traffic Classes (TC) : 4							
Traffic Class (TC)	Traffic Type (TT)	TC Description (TD)	TC Priority Map (TP)	TC Bandwidth (%) Min	Max	TC Flow Control (TF)	TC Scheduler (TS)
0	BE	Best Effort	0, 1	27	100	Lossy	ETS
1	CA	Critical Applications	3, 2	40	100	Lossy	ETS
2	VO	Voice (<10ms latency)	5, 4	33	100	Loss Less	ETS
3	NC	Network Control	7, 6	0	100	Lossy	Strict Priority

DCB Profile 3

Profile Control :: DCBX Configuration

PFC :: Enable TLV : [*Yes/No]   Recommendation :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]   Recommended Profile : [DCB_Def_Profile_1/NULL]
    Capability : 8

                            Application :: Enable TLV : [Yes/*No]
ETS :: Enable TLV : [*Yes/No]   Defense Mode :: [*Enable/Disable]
    Willing : [*Yes/No]   Default MTU :: [9216]
    Max TC : 3

Profile Data :: Max Traffic Classes (TC) : 5

Traffic Class (TC)	Traffic Type (TT)	TC Description (TD)	TC Priority Map (TP)	TC Bandwidth (%) Min	Max	TC Flow Control (TF)	TC Scheduler (TS)
0	BE	Best Effort	0, 1	27	100	Lossy	ETS
1	CA	Critical Applications	3, 2	40	100	Lossy	ETS
2	VO	Voice (<10ms latency)	5, 4	33	100	Loss Less	ETS
3	IC	Internetwork Control	6	0	100	Lossy	Strict Priority
4	NC	Network Control	7	0	100	Lossy	Strict Priority

DCB Profile 4

Profile Control :: DCBX Configuration

PFC :: Enable TLV : [*Yes/No]   Recommendation :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]   Recommended Profile : [DCB_Def_Profile_1/NULL]
    Capability : 8

                            Application :: Enable TLV : [Yes/*No]
ETS :: Enable TLV : [*Yes/No]   Defense Mode :: [*Enable/Disable]
    Willing : [*Yes/No]   Default MTU :: [9216]
    Max TC : 3

Profile Data :: Max Traffic Classes (TC) : 6

Traffic Class (TC)	Traffic Type (TT)	TC Description (TD)	TC Priority Map (TP)	TC Bandwidth (%) Min	Max	TC Flow Control (TF)	TC Scheduler (TS)
0	BK	Background	1	7	100	Lossy	ETS
1	BE	Best Effort	0	20	100	Lossy	ETS
2	CA	Critical Applications	3, 2	40	100	Lossy	ETS
3	VO	Voice (<10ms latency)	5, 4	33	100	Loss Less	ETS
4	IC	Internetwork Control	6	0	100	Lossy	Strict Priority
5	NC	Network Control	7	0	100	Lossy	Strict Priority

DCB Profile 5							
Profile Control :: DCBX Configuration							

PFC :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Capability : 8
 	
ETS :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Max TC : 3
				Recommendation :: Enable TLV : [*Yes/No]
				Recommended Profile : [DCB_Def_Profile_1/NULL]
				
				Application :: Enable TLV : [Yes/*No]
				Defense Mode :: [*Enable/Disable]
				Default MTU :: [9216]			

Profile Data :: Max Traffic Classes (TC) : 7							
Traffic Class (TC)	Traffic Type (TT)	TC Description (TD)	TC Priority Map (TP)	TC Bandwidth (%) Min	Max	TC Flow Control (TF)	TC Scheduler (TS)
0	BK	Background	1	7	100	Lossy	ETS
1	BE	Best Effort	0	20	100	Lossy	ETS
2	EE	Excellent Effort	2	25	100	Lossy	ETS
3	CA	Critical Applications	3	15	100	Lossy	ETS
4	VO	Voice (<10ms latency)	5, 4	33	100	Loss Less	ETS
5	IC	Internetwork Control	6	0	100	Lossy	Strict Priority
6	NC	Network Control	7	0	100	Lossy	Strict Priority

DCB Profile 6							
Profile Control :: DCBX Configuration							

PFC :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Capability : 8
 	
ETS :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Max TC : 3
				Recommendation :: Enable TLV : [*Yes/No]
				Recommended Profile : [DCB_Def_Profile_1/NULL]
				
				Application :: Enable TLV : [Yes/*No]
				Defense Mode :: [*Enable/Disable]
				Default MTU :: [9216]			

Profile Data :: Max Traffic Classes (TC) : 8							
Traffic Class (TC)	Traffic Type (TT)	TC Description (TD)	TC Priority Map (TP)	TC Bandwidth (%) Min	Max	TC Flow Control (TF)	TC Scheduler (TS)
0	BK	Background	1	7	100	Lossy	ETS
1	BE	Best Effort	0	20	100	Lossy	ETS
2	EE	Excellent Effort	2	15	100	Lossy	ETS
3	CA	Critical Applications	3	25	100	Lossy	ETS
4	VI	Video (<100ms latency)	4	15	100	Loss Less	ETS
5	VO	Voice (<10ms latency)	5	18	100	Loss Less	ETS
6	IC	Internetwork Control	6	0	100	Lossy	Strict Priority
7	NC	Network Control	7	0	100	Lossy	Strict Priority

DCB Profile 7

Profile Control :: DCBX Configuration

PFC :: Enable TLV : [*Yes/No]   Recommendation :: Enable TLV : [*Yes/No]
     Willing : [*Yes/No]   Recommended Profile : [DCB_Def_Profile_1/NULL]
     Capability : 8

                                    Application :: Enable TLV : [Yes/*No]
ETS :: Enable TLV : [*Yes/No]   Defense Mode :: [*Enable/Disable]
     Willing : [*Yes/No]   Default MTU :: [9216]
     Max TC : 3

Profile Data :: Max Traffic Classes (TC) : 8

Traffic Class (TC)	Traffic Type (TT)	TC Description (TD)	TC Priority Map (TP)	TC Bandwidth (%) Min	Max	TC Flow Control (TF)	TC Scheduler (TS)
0	BE	Best Effort	0	0	100	Loss Less	Strict Priority
1	BK	Background	1	0	100	Loss Less	Strict Priority
2	EE	Excellent Effort	2	0	100	Loss Less	Strict Priority
3	CA	Critical Applications	3	0	100	Loss Less	Strict Priority
4	VI	Video (<100ms latency)	4	0	100	Loss Less	Strict Priority
5	VO	Voice (<10ms latency)	5	0	100	Loss Less	Strict Priority
6	IC	Internetwork Control	6	0	100	Loss Less	Strict Priority
7	NC	Network Control	7	0	100	Loss Less	Strict Priority

DCB Profile 8 (Default Profile)

Profile Control :: DCBX Configuration

PFC :: Enable TLV : [*Yes/No]   Recommendation :: Enable TLV : [*Yes/No]
     Willing : [*Yes/No]   Recommended Profile : [DCB_Def_Profile_1/NULL]
     Capability : 8

                                      Application :: Enable TLV : [Yes/*No]
ETS :: Enable TLV : [*Yes/No]   Defense Mode :: [*Enable/Disable]
     Willing : [*Yes/No]   Default MTU :: [9216]
     Max TC : 3

Profile Data :: Max Traffic Classes (TC) : 8

Traffic Class (TC)	Traffic Type (TT)	TC Description (TD)	TC Priority Map (TP)	TC Bandwidth (%) Min	Max	TC Flow Control (TF)	TC Scheduler (TS)
0	BE	Best Effort	0	0	100	Lossy	Strict Priority
1	BK	Background	1	0	100	Lossy	Strict Priority
2	EE	Excellent Effort	2	0	100	Lossy	Strict Priority
3	CA	Critical Applications	3	0	100	Lossy	Strict Priority
4	VI	Video (<100ms latency)	4	0	100	Lossy	Strict Priority
5	VO	Voice (<10ms latency)	5	0	100	Lossy	Strict Priority
6	IC	Internetwork Control	6	0	100	Lossy	Strict Priority
7	NC	Network Control	7	0	100	Lossy	Strict Priority

DCB Profile 9

Profile Control :: DCBX Configuration

PFC :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Capability : 8

ETS :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Max TC : 3

Recommendation :: Enable TLV : [*Yes/No]
Recommended Profile : [DCB_Def_Profile_1/NULL]

Application :: Enable TLV : [Yes/*No]
Defense Mode :: [*Enable/Disable]
Default MTU :: [9216]

Profile Data :: Max Traffic Classes (TC) : 8

Traffic Class (TC)	Traffic Type (TT)	TC Description (TD)	TC Priority Map (TP)	TC Bandwidth (%) Min	Max	TC Flow Control (TF)	TC Scheduler (TS)
0	BK	Background	1	12	100	Loss Less	ETS
1	BE	Best Effort	0	12	100	Loss Less	ETS
2	EE	Excellent Effort	2	12	100	Loss Less	ETS
3	CA	Critical Applications	3	12	100	Loss Less	ETS
4	VI	Video (<100ms latency)	4	13	100	Loss Less	ETS
5	VO	Voice (<10ms latency)	5	13	100	Loss Less	ETS
6	IC	Internetwork Control	6	13	100	Loss Less	ETS
7	NC	Network Control	7	13	100	Loss Less	ETS

DCB Profile 10

Profile Control :: DCBX Configuration

PFC :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Capability : 8

ETS :: Enable TLV : [*Yes/No]
    Willing : [*Yes/No]
    Max TC : 3

Recommendation :: Enable TLV : [*Yes/No]
Recommended Profile : [DCB_Def_Profile_1/NULL]

Application :: Enable TLV : [Yes/*No]
Defense Mode :: [*Enable/Disable]
Default MTU :: [9216]

Profile Data :: Max Traffic Classes (TC) : 8

Traffic Class (TC)	Traffic Type (TT)	TC Description (TD)	TC Priority Map (TP)	TC Bandwidth (%) Min	Max	TC Flow Control (TF)	TC Scheduler (TS)
0	BK	Background	1	12	100	Lossy	ETS
1	BE	Best Effort	0	12	100	Lossy	ETS
2	EE	Excellent Effort	2	12	100	Lossy	ETS
3	CA	Critical Applications	3	12	100	Lossy	ETS
4	VI	Video (<100ms latency)	4	13	100	Lossy	ETS
5	VO	Voice (<10ms latency)	5	13	100	Lossy	ETS
6	IC	Internetwork Control	6	13	100	Lossy	ETS
7	NC	Network Control	7	13	100	Lossy	ETS

DCB Profile 11							
Profile Control :: DCBX Configuration							
PFC :: Enable TLV : [*Yes/No] Willing : [*Yes/No] Capability : 8 ETS :: Enable TLV : [*Yes/No] Willing : [*Yes/No] Max TC : 3				Recommendation :: Enable TLV : [*Yes/No] Recommended Profile : [DCB_Def_Profile_1/NULL] Application :: Enable TLV : [Yes/*No] Defense Mode :: [*Enable/Disable] Default MTU :: [9216]			
Profile Data :: Max Traffic Classes (TC) : 8							
Traffic Class (TC)	Traffic Type (TT)	TC Description (TD)	TC Priority Map (TP)	TC Bandwidth (%) Min Max		TC Flow Control (TF)	TC Scheduler (TS)
0	BE	Best Effort	0	12	100	Loss Less	ETS
1	BK	Background	1	12	100	Loss Less	ETS
2	EE	Excellent Effort	2	12	100	Loss Less	ETS
3	CA	Critical Applications	3	12	100	Loss Less	ETS
4	VI	Video (<100ms latency)	4	13	100	Loss Less	ETS
5	VO	Voice (<10ms latency)	5	13	100	Loss Less	ETS
6	IC	Internetwork Control	6	13	100	Loss Less	ETS
7	NC	Network Control	7	13	100	Loss Less	ETS

The Windows driver for an Intel NIC has a fixed number of traffic classes (eight) and the priority to TC mapping uses a linear mapping. To support ETS BW allocations with eight traffic classes and linear priority to traffic class mapping, use DCB profile 11 (DCP 11).

Configuring DCB Profiles

The default DCB profile (DCP 8) is automatically assigned to each QSet instance when a port goes active or a port joins a LAG. It is only necessary to assign a different profile if DCP 8 attributes are not sufficient.

Consider the following when configuring DCB profiles:

- DCB profiles 1–11 are predefined profiles that are not modifiable and cannot be deleted from the switch configuration.

- Creating a custom profile is allowed by importing one of the pre-defined profiles into a new profile ID between 12 and 128, then modifying the new profile attributes as necessary.

- There is only one DCP assigned to each QSet instance and only one QSet instance per port or link aggregate (LAG). However, a LAG may show multiple QSet instances, one for each port that is a member of the LAG.

- When a port leaves a LAG, the default DCP 8 profile is associated with the QSet instance for that port. In other words, if the QSet instance for a port was associated with DCP 4 when the port joined the LAG, the port is associated with DCP 8 when it leaves the LAG.

Creating a Custom Profile

The **qos qsp dcb import** command is used to create a custom profile using one of the 11 pre-defined DCB profiles as a template. For example, the following command creates profile 20 (DCP 20) using pre-defined DCP 5 as the template for the new profile:

```
-> qos qsp dcb 20 import qsp dcb 5
```

Once the custom profile is created, the following traffic class attributes for the profile are configurable using the **qos qsp dcb tc** command:

TC Attribute	Command Parameters	Description
PFC flow control	**pfc flow-type nll | ll**	Changes the traffic class to lossy or loss less.
PFC link delay	**pfc link-delay** *10-100*	Sets the actual headroom value for the traffic class. Setting an incorrect value for this command may result in traffic loss. In most cases, the profile default value of 0 for lossy and 52 for lossless is sufficient.
Bandwidth	**min-bw | max-bw**	Configures the minimum and maximum bandwidth for the traffic class.
Recommended bandwidth	**recommended bw**	Configures the recommended minimum bandwidth for the traffic class.

Guidelines for Modifying Custom Profiles

1 Changing the priority-to-TC mapping is not allowed.

2 Changing the TC scheduler type (SP, ETS) is not allowed.

3 Enabling or disabling PFC is allowed on any TC in the custom profile. The PFC status applies to all priorities within the specific TC.

4 Changing Weights (Min Rate) on any of the ETS TC is allowed, with ?ETS_TC_Weights < = 100% Available_PR (after SP allocation).

5 Changing PIR (shaper)/Max Rate is allowed on all TC Classes/Types (essential for high Priority SP (runaway) classes).

6 When a custom profile is modified, the changes are applied to all ports that are associated with that custom profile. To apply specific changes to a single port (QSet instance), import a custom or default profile into a new custom profile, make the necessary changes, then apply the new custom profile to the port.

Changing the Profile Assignment

To assign a different profile to a specific QSet instance (QSI), use the **qos qsi qsp dcb** command. For example, the following commands assign DCP 2 to port 1/2 and DCP 3 to ports 2/1-10 and linkagg 5:

```
-> qos qsi port 1/2 qsp dcb 2
-> qos qsi port 2/1-10 qsp dcb 3
-> qos qsi linkagg 5 qsp dcb 3
```

To view the DCB profile configuration for the switch, use the **show qos qsp dcb** command. For example:

```
-> show qos qsp dcb

Legends: Prio TC Map:
            Represents the priority to traffic class mapping;
            begins with priority 0 on the left and displays the
            traffic class it belongs to.

                                     ETS
                   Priority PFC Max Template-DCP      802.3x
#    Name          TC Map   Cap TC  #   Name          Pause-Ready
---+------------+--------+---+---+---+------------+-----------
1    dcp-1         00001122 8   8   1   dcp-1         No
2    dcp-2         00112233 8   8   2   dcp-2         No
3    dcp-3         00112234 8   8   3   dcp-3         No
4    dcp-4         10223345 8   8   4   dcp-4         No
5    dcp-5         10234456 8   8   5   dcp-5         No
6    dcp-6         10234567 8   8   6   dcp-6         No
7    dcp-7         01234567 8   8   7   dcp-7         No
8    dcp-8         01234567 8   8   8   dcp-8         No
9    dcp-9         10234567 8   8   9   dcp-9         No
10   dcp-10        10234567 8   8   10  dcp-10        No
20   dcp-20        10234567 8   8   10  dcp-10        No
```

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

Configuring Support for Legacy Pause Frames

Configuring support for legacy PAUSE frames on a DCB OmniSwitch is useful when the switch needs to interoperate with a server that does not support PFC. Support for legacy PAUSE frames is configurable by creating a custom profile with the 802.3x pause attribute enabled.

1 Create a custom profile based on any pre-defined profile and enable PAUSE. For example:

```
-> qos qsp dcb 12 import qsp dcb dcp-9 802.3x-pause
```

2 Disable PFC TLV and PFC willing on the port that will support the PAUSE frames. For example:

```
-> qos qsi port 1/1/5 dcb dcbx pfc tlv disable
-> qos qsi port 1/1/5 dcb dcbx pfc willing no
```

3 Enable pause on the port. For example:

```
-> interfaces 1/1/5 pause tx-and-rx
```

4 Apply the custom profile, "dcp-12" to the port. For example:

```
-> qos qsi port 1/1/5 qsp dcb 12
```

When this type of custom profile is applied on an ingress port, a legacy PAUSE frame is sent on the ingress port back to the server during congestion on the egress.

Configuring DCBX Port Parameters

By default, the Data Center Bridging Exchange (DCBX) protocol is enabled on all switch ports when the switch boots up with the OmniSwitch Data Center software license. To disable DCBX for a port or link aggregate, use the **qos qsi dcb dcbx admin-state** command. For example:

```
-> qos qsi port 1/10 dcb dcbx admin-state disable
-> qos qsi linkagg 2 dcb dcbx admin-state disable
```

In addition to configuring the status of DCBX on a port, the following DCBX port attributes are also configurable using the **qos qsi dcb dcbx pfc** and **qos qsi dcb dcbx ets** commands:

- The "willing" bit setting (Yes or No) for PFC and ETS. This parameter is set to Yes, which means that when a profile configuration mismatch occurs between two directly connected devices:
  – The PFC settings are resolved based on the settings of the device with the lowest MAC address.
  – The ETS settings from the other device are accepted by each device.
- The status of configuration TLV transmission for PFC and ETS (enabled by default).
- The status of the recommended TLV for ETS (enabled by default).

To verify the DCBX port configuration, use the **show qos qsi dcb dcbx** command. For example:

```
-> show qos qsi dcb dcbx
                                                    ETS
                  DCBX  Stats PFC     PFC PFC Cfg Reco ETS  App
Port   DCP Name   Admin Admin Defense TLV Will TLV TLV  Will TLV
-------+---+------------+-----+-----+-------+---+----+---+----+----+---
1/1    8   dcp-8    Ena   Dis   Ena     Ena Yes  Ena Ena  Yes  Dis
1/2    8   dcp-8    Ena   Dis   Ena     Ena Yes  Ena Ena  Yes  Dis
1/3    8   dcp-8    Ena   Dis   Ena     Ena Yes  Ena Ena  Yes  Dis
1/4    8   dcp-8    Ena   Dis   Ena     Ena Yes  Ena Ena  Yes  Dis
1/5    8   dcp-8    Ena   Dis   Ena     Ena Yes  Ena Ena  Yes  Dis
1/6    8   dcp-8    Ena   Dis   Ena     Ena Yes  Ena Ena  Yes  Dis
1/7    8   dcp-8    Ena   Dis   Ena     Ena Yes  Ena Ena  Yes  Dis
1/8    8   dcp-8    Ena   Dis   Ena     Ena Yes  Ena Ena  Yes  Dis
10     8   dcp-8    Ena   Dis   Ena     Ena Yes  Ena Ena  Yes  Dis
vfl    8   dcp-8    Ena   Dis   Ena     Ena Yes  Ena Ena  Yes  Dis
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the **qos qsi dcb dcbx** and related **show** commands.

Multicast and Unicast Traffic Distribution

The following Class of Service (CoS) model for unicast and multicast traffic is applied when either the default QSet profile (QSP 1) or the default DCB profile (DCP 8) is the active profile for the port.

Cos 0 - Lower Priority MC (0-3)   = 10

Cos 1 - Higher Priority MC (4-7)   = 52

Cos 3 - All Other Unicast UC(0-7) =108

Cos 7 - CPU Generated Packets   = 127 (maximum weight)

For example:

- When sending two streams of 100% MC Lower Priority and 100% MC Higher Priority, the distribution should be 10 and 50 packets, which is approximately 17% of Lower Priority MC and 83% of Higher Priority.

- When sending Lower Priority MC 100% and UC 100%, the distribution is 9% of MC and 91% of UC.

- When sending Higher Priority MC 100% and UC 100%, the distribution is 32% of MC and 68% of UC.

Non-Default Profile

The CoS model implemented also applies for non-default QSet profiles (QSP 2–4), except on the OmniSwitch 6900. The multicast and unicast queue mapping for non-default QSet profiles (QSP 2–4) and non-default DCB profiles (DCP 1–7, 9–128) on the OmniSwitch 6900, is as follows:

Strict Priority Profiles (for example, DCP 7)

Queues	Priority	Precedence
UC7	7	Highest
MC3	7, 6	
UC6	6	
UC5	5	
MC2	5, 4	
UC4	4	
UC3	3	
MC1	3, 2	
UC2	2	
UC1	1	
MC0	1, 0	
UC0	0	Lowest

Weighted Round Robin (WRR) Profiles

Queues	Priority	Weight
UC7	7	W7
MC3	7, 6	Avg(W7,W6)
UC6	6	W6
UC5	5	W5
MC2	5, 4	Avg(W5,W4)
UC4	4	W4
UC3	3	W3
MC1	3, 2	Avg(W3,W2)
UC2	2	W2
UC1	1	W1
MC0	1, 0	Avg(W1,W0)
UC0	0	W0

Note: Wn = Weight of UCn

Avg(Wn,Wm) = Average of Weights of UCn & UCm

Profile with a Mix of Strict Priority and WRR

Unicast queues configured as Strict Priority will inherit behavior from the Strict Priority model, and unicast queues configured as WRR will inherit behavior from the WRR model. Multicast queues will always follow the behavior that the corresponding unicast queues are following. For example:

- If UC7 and UC6 are Strict Priority, then the MC3 (priority 6 and 7) will also use Strict Priority.

- If UC7 and UC6 are Weighted Round Robin, then MC3 (priority 6 and 7) will also use Weighted Round Robin. The weight of MC3 will be the average of the weights for UC6 and UC7.

For DCB profile ETS behavior, where a Traffic Class (TC) can have more than one priority, multicast queues will follow the corresponding unicast queue behavior. For example:

DCB Profile 1:

TC 0 (Priority 0 -3)

TC 1 (Priority 4 -5)

TC 2 (Priority 6 -7)

TC 0 has UC0 through UC3 in Round Robin, so MC0 (priority 0 and 1) and MC1 (priority 2 and 3) will also participate in the Round Robin behavior of TC 0.

Multicast and Unicast Traffic Distribution for the OmniSwitch 6900-Q32 and OmniSwitch 6900-X72

The multicast and unicast queue mapping for the OmniSwitch 6900-Q32 and OmniSwitch 6900-X72 is as follows:

Queues	Priority	Precedence
UC7, MC7	7	Highest
UC6, MC6	6	
UC5, MC5	5	
UC4, MC4	4	
UC3, MC3	3	
UC2, MC2	2	
UC1, MC1	1	
UC0, MC0	0	Lowest

There are additional multicast queues in the OmniSwitch 6900-Q32 and OmniSwitch 6900-X72. As a result, traffic distribution is based on the priority value regardless of whether the traffic is multicast or unicast. In other words, multicast traffic with priority 1 and unicast with priority 1 are given equal distribution.

Multicast Source PFC on OmniSwitch 6900

Ingress admission control on the OmniSwitch 6900 does not distinguish between unicast and multicast traffic. Therefore, a multicast source connected to a port which is PFC aware will react to congestion thereby pausing transmission. This will affect multicast hosts not in the congestion path.

When a multicast source is attached to a port on a OmniSwitch 6900, make sure that PFC is not enabled for that particular priority on the ingress. This can be done by configuring the port to use DCP 8 (all priorities are lossless) or for instance, DCB-1 (priority 4 and 5 are lossless, so multicast may be sent at any other priority other than priority 4 or 5).

If multicast sources are configured to react to PFC, it will affect subscribers not in the congestion path.

Verifying the DCB Configuration

Displaying the Data Center Bridging (DCB) configuration is helpful to verify the actual configuration on each switch in the data center mesh topology. To display information about the DCB configuration, use the **show** commands listed in this section.

show qos qsp dcb	Displays the configured DCB profiles and the traffic classes associated with the DCB profile.
show qos qsi dcb dcbx	Displays the Data Center Bridging Exchange (DCBX) port configuration and status.
show qos qsi dcb ets	Displays the DCB Enhanced Transmission Selection (ETS) port configuration and status.
show qos qsi dcbx pfc	Displays the DCB Priority-based Flow Control (PFC) port configuration and status.
show qos pfc-lossless-usage	Displays the PFC lossless priority usage for the switch.
show qos qsi dcb pfc stats	Displays PFC port statistics.
show configuration snapshot vfc	Displays the current running configuration for DCB, QSP, PFC, and ETS.

3   Configuring Shortest Path Bridging

The OmniSwitch supports Shortest Path Bridging MAC (SPBM), as defined in the IEEE 802.1aq standard. SPBM uses the Provider Backbone Bridge (PBB) network model to encapsulate (using IEEE 802.1ah headers) and tunnel customer traffic through the network backbone. The shortest path trees upon which the PBB network infrastructure operates are determined using a version of the Intermediate System-to-Intermediate System (IS-IS) link state protocol that supports TLV extensions for SPB (ISIS-SPB).

Incorporating SPBM into the data center infrastructure provides the following benefits:

- Transparently extends Layer 2 connections (VLAN segments) across a large virtual service Layer 2 backbone network.

- Maintains a loop-free network while providing efficient use of available bandwidth, especially in a mesh topology. All connections between all switches in the topology remain active (no blocking of redundant links).

- A shortest path is automatically calculated between each bridge and every other bridge in the data center mesh, resulting in low latency and sub-second convergence times needed to support critical data center bridging requirements.

- Can process a large number of customer MAC addresses without overrunning provider network resources. Customer MAC addresses are only learned on Backbone Edge Bridges (BEB), where customer traffic is then encapsulated and tunneled through the network core infrastructure. Backbone Core Bridges (BCB) do not have to learn any customer MAC addresses.

- Provides a clear separation of customer traffic (between different customers and between the provider network domain). Entry points for customer traffic are clearly defined on the participating BEBs. Customer traffic is identified and associated with a specific service instance bound to the PBB infrastructure.

- Integration with Virtual Machine Network Profiles (vNPs) to support virtual machine (VM) discovery and mobility.

In This Chapter

This chapter provides an overview about how Shortest Path Bridging MAC (SPBM) works and how to configure SPBM through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

This chapter includes the following topics:

SPBM Parameter Defaults

Parameter Description	Command	Default
ISIS-SPB status for the switch.	**spb isis admin-state**	Disabled
Equal Cost Tree (ECT) ID number for the backbone VLAN (BVLAN).	**spb isis bvlan ect-id**	1 or next available ECT ID number on the local switch.
Control BVLAN for the switch.	**spb isis control-bvlan**	None
The BVLAN tandem multicast mode (only applies to associated SPB services running in tandem mode).	**spb isis bvlan tandem-multicast-mode**	Source and Group (S, G)
Priority value for the ISIS-SPB instance.	**spb isis bridge-priority**	32768
Wait time intervals, in milliseconds, for shortest path first (SPF) calculations.	**spb isis spf-wait**	maximum wait: 1000 initial wait : 100 second wait : 300
Wait time intervals, in milliseconds, for link state PDU (LSP) transmissions.	**spb isis lsp-wait**	maximum wait: 1000 initial wait : 0 second wait : 300
Graceful restart status for the switch.	**spb isis graceful-restart**	Enabled
Graceful restart helper status for the switch.	**spb isis graceful-restart helper**	Enabled

SPBM Interface Defaults

Parameter Description	Command	Default
SPB interface status	**spb isis interface**	Disabled
SPB interface time interval between each hello packet transmission.	**spb isis interface hello-interval**	9 seconds
SPB interface hello multiplier used to determine hello packet hold time.	**spb isis interface hello-multiplier**	3
SPB interface link cost to reach the peer bridge.	**spb isis interface metric**	10

SPBM Service Defaults

By default, there are no SPBM service components configured for the switch. However, when a service is created, the following default values apply:

Parameter Description	Command	Default
SPB service administrative status.	**service admin-state**	Disabled
SPB service multicast replication mode.	**service multicast-mode**	Head-end
SPB service VLAN translation.	**service vlan-xlation**	Disabled
SPB service maximum transmission unit (MTU) value.	Not configurable at this time	9194
SPB service statistics collection.	**service stats**	Disabled
SPB service description.	**service description**	None.
Default profile automatically applied to access ports.	**service access l2profile**	def-access-profile
Layer 2 profile that specifies how control packets are processed on service access ports.	**service l2profile**	def-access-profile: STP, GVRP, MVRP = tunnel 802.3ad = peer 802.1x, 802.1ab, AMAP = drop CSCO PDU, VLAN, uplink = drop
VLAN translation for the service access port.	**service access vlan-xlation**	Disabled
Service access point (SAP) administrative status.	**service sap admin-state**	Enabled
SAP encapsulation.	**service sap**	0 (untagged traffic).
SAP trust mode.	**service sap trusted**	Trusted
SAP statistics collection.	**service sap stats**	Disabled
SAP description.	**service sap description**	None

Shortest Path Bridging Overview

The OmniSwitch implementation of Shortest Path Bridging (SPB) supports SPB MAC (SPBM) as defined in the IEEE 802.1aq standard. SPBM is defined for use in Provider Backbone Bridge (PBB) networks as specified in the IEEE 802.1ah standard.

SPBM provides a mechanism to automatically define a shortest path tree (SPT) bridging configuration through a Layer 2 Ethernet network. SPBM Ethernet services use this configuration to encapsulate and tunnel data through the PBB network. The following main components of the OmniSwitch implementation of SPBM provide this type of functionality:

- **ISIS-SPB**—A version of the Intermediate to Intermediate System (IS-IS) link state protocol that supports SPB TLV extensions. SPBM uses ISIS-SPB to build sets of symmetric shortest path trees (SPTs) between any SPB switch.

- **Provider Backbone Bridge (PBB) IEEE 802.1ah**— Defines a MAC-in-MAC data encapsulation path for PBB networks that is supported by SPBM.

- **Provider Backbone Bridge Network (PBBN)**—A network comprised of Backbone Edge Bridges (BEBs) and Backbone Core Bridges (BCB) that is used to interconnect Provider Bridge Networks (PBN) with other networks.

- **Backbone Edge Bridge (BEB)**—An SPB switch positioned at the edge of the PBB network that learns and encapsulates (adds an 802.1ah backbone header to) customer frames for transport across the backbone network. The BEB interconnects the customer network space with PBB network space.

- **Backbone Core Bridge (BCB)**—An SPB node that resides inside the PBB network core. The BCB employs the same BVLAN on two or more network ports. This BVLAN does not terminate on the switch itself; traffic ingressing on an SPB network port is switched to other SPB network ports. As a result, the BCB does not have to learn any of the customer MAC addresses. It mainly serves as a transit bridge for the PBB network.

- **SPBM Service**—An OmniSwitch Service Manager service configured on the BEBs. Each service maps to a service instance identifier (I-SID) which is bound to a backbone VLAN. One backbone VLAN can accommodate multiple I-SIDs.

- **Backbone VLAN (BVLAN)**—A VLAN that serves as a transport VLAN for the SPBM service instances and to connect SPB bridges together through SPT sets. Unlike standard VLANs, BVLANs do not learn source MAC addresses or flood unknown destination or multicast frames. Instead, BVLANs only forward on the basis of the forwarding database (FDB) as populated by the ISIS-SPB protocol.

The following diagram shows how SPBM uses the above components to tunnel customer traffic through a Provider Backbone Bridge Network:
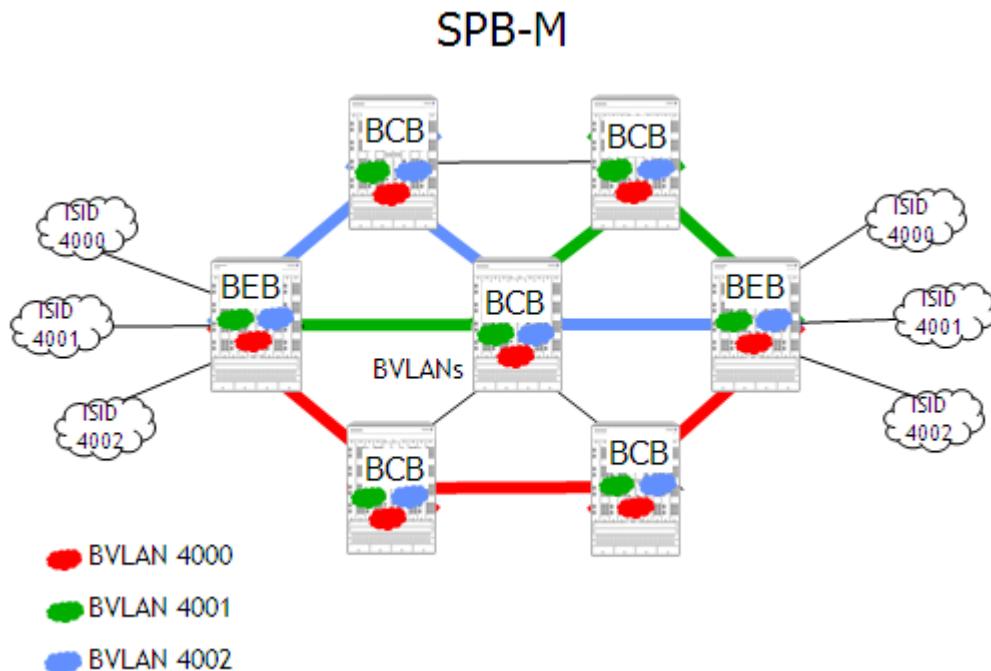
Figure 1: SPBM Network Components

In this network,

- The BEBs are SPBM capable (ISIS-SPB configured and enabled) and form a shortest path bridging network that also includes the SPBM capable Backbone Core Bridges (BCBs).

- Each bridge calculates a shortest path tree (SPT) for each BVLAN with itself as the root of each tree.

- SPB Ethernet service instances identified by I-SIDs are created on each BEB. Each I-SID is associated with a BVLAN ID. The BVLAN is configured on each bridge (BEB and BCB) in the backbone network. However, the I-SID itself and the I-SID association with the BVLAN is only configured on each BEB that will service customer traffic.

- A Service Access Point (SAP) is configured on each BEB to identify the access port on which customer traffic will enter the PBBN, the SPB service instance that will tunnel the traffic through the network, and the type of customer traffic to forward (for example, only specific CVLAN IDs, untagged traffic only, or all tagged traffic). Basically, the SAP binds access ports and the specified customer traffic received on those ports to the service.

- Layer 2 traffic from the connected edge networks enters the BEBs through access ports. The SAP configuration on the receiving access port is applied to classify which frames are mapped to which services, if any.

- Classified traffic is then encapsulated into 802.1ah frames by the BEB before the frames are transmitted through the backbone network.

- The 802.1ah encapsulated frames are forwarded on the shortest path through the entire PBBN to reach the intended destination BEB. The BCBs switch traffic based on the destination backbone MAC address (BMAC)—bridge MAC address of the BEB—provided in the 802.1ah header and do not process any I-SID information in the frame.

SPBM Shortest Path Trees

The shortest path between two points is a straight line. Shortest Path Bridging (SPB) implements frame forwarding on the shortest path between any two bridges in an Ethernet network. The shortest path trees (SPTs) calculated by SPB provide the shortest and most efficient path to and from the intended destination. SPTs are formed along the direct, straight-line links between switches to make up an overall path through the topology that provides a robust, efficient direction for network traffic to travel.

The SPBM network topology consists of two layers:

- **The backbone infrastructure (control plane) layer**. ISIS-SPB builds the backbone layer by defining loop-free, shortest path trees (SPTs) through the backbone network.

- **The services (data plane) layer**. The service layer is based on the Provider Backbone Bridging (PBB) framework as defined in the IEEE 802.1ah standard. SPBM supports the 802.1ah MAC-in-MAC method for data encapsulation. SPBM services transport the encapsulated traffic over the ISIS-SPB infrastructure. (See "SPB Services" on page 3-11 for more information).

This section contains an example of ISIS-SPB operations in a small SPBM network. In addition to describing how shortest path trees are created in the BVLAN domain, the flow of unicast and multicast traffic through the network, this example also shows the benefits of using SPB over Spanning Tree for VLAN traffic distribution.

Spanning Tree

The following diagram shows an example Provider Backbone Bridge (PBB) network with a single backbone VLAN using the Spanning Tree protocol for network loop protection with the same path cost on all links:
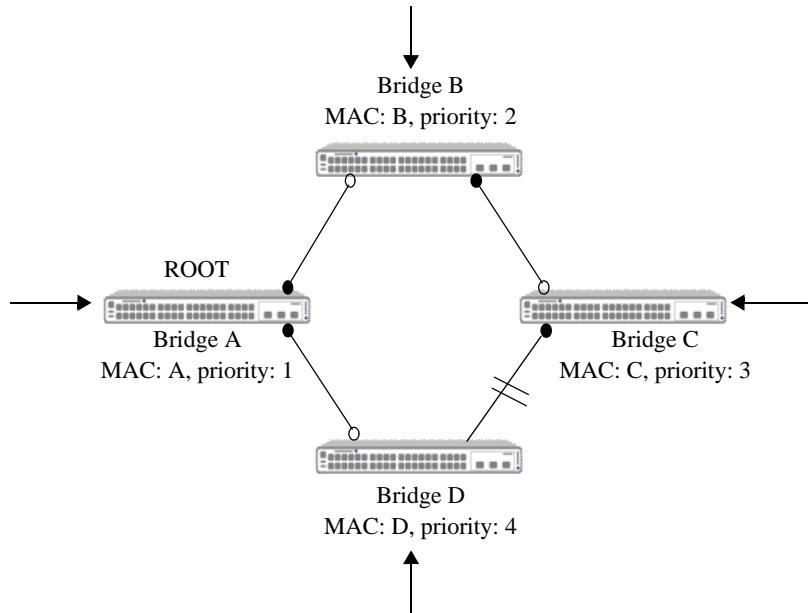
Figure 2: Spanning Tree Topology

In this example, Bridge A is the Root bridge. As a result, customer traffic entering Bridge A would always use the shortest path to reach every other bridge in the network. However, traffic entering Bridge D that is destined for Bridge C must traverse the path through Bridge A to reach Bridge C, even though Bridge D is directly connected to Bridge C. Clearly the path from Bridge D to Bridge C is not the shortest path in this case.
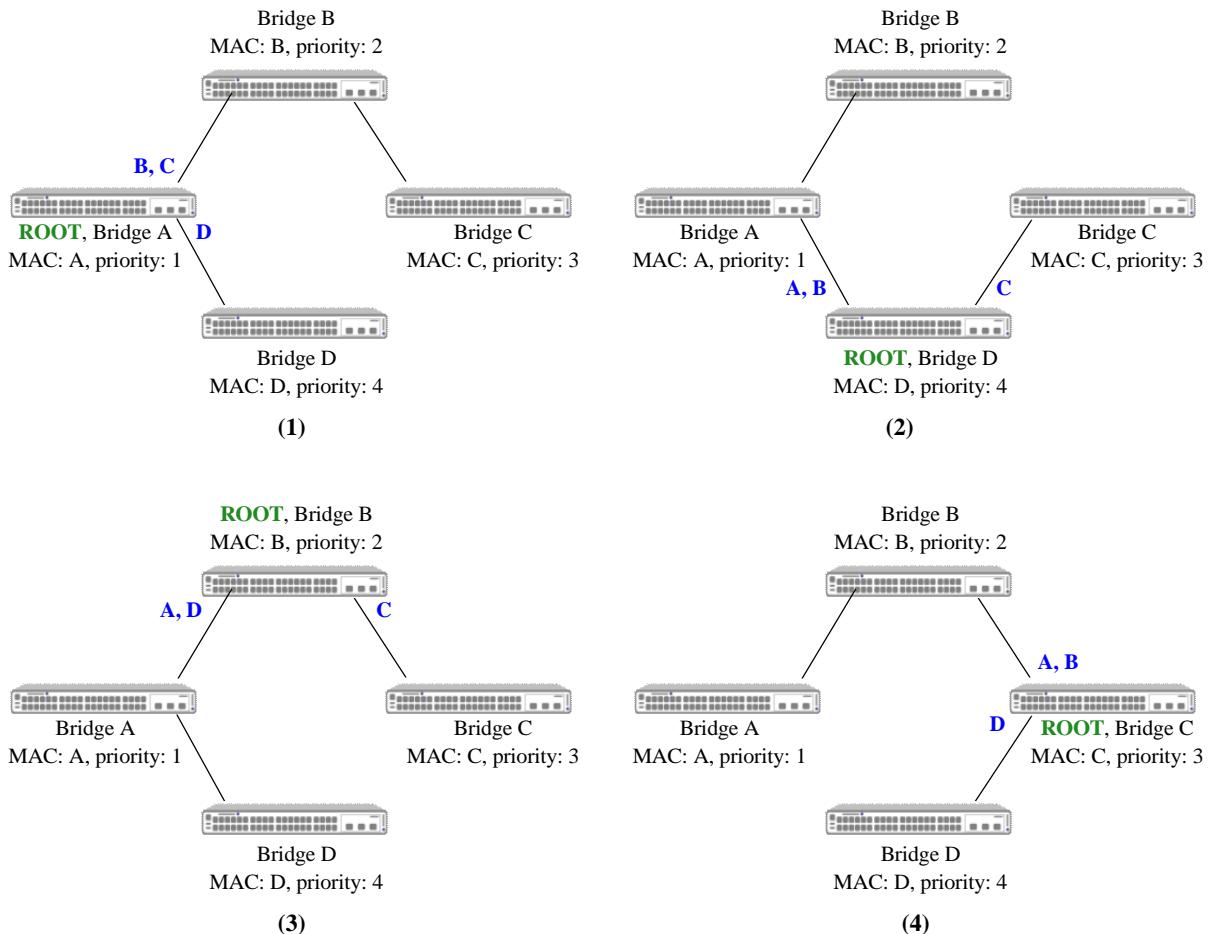
ISIS-SPB

The IEEE 802.1aq standard for SPB specifies the use of the IS-IS link state protocol instead of Spanning Tree to form sets of shortest path trees through the network. When SPB is used, each bridge is the Root for all traffic entering that bridge. As a result, each bridge can provide the shortest path to every other bridge in the network.

The bridging methodology needed to allow each bridge to serve as its own root bridge is enforced through the use of SPB BVLANs. This type of VLAN does not learn customer MAC addresses or flood unknown unicast and multicast traffic. In addition, network loops are mitigated through strict ingress checks based on the source MAC address of frames received on the BVLAN (frames received from an unexpected source are discarded).

SPBM uses an extended version of the IS-IS protocol that supports SPB (ISIS-SPB) to calculate the SPBM network topology. In addition, the learning and propagation of source MAC addresses is handled through the ISIS-SPB control plane, instead of through the data plane.

When calculating the SPBM network topology, ISIS-SPB must meet Layer 2 requirements to create congruent and symmetric paths. To do this, SPBM supports 16 predefined Equal Cost Tree (ECT) algorithms to break ties when two or more equal cost paths to the same destination are discovered. The same ECT algorithm is configured for the same BVLAN ID on each SPB switch in the network to ensure congruent, symmetric paths for the service traffic bound to that BVLAN.

Basically, to create a unicast tree, SPBM simply computes the shortest path from every bridge with each bridge serving as the Root (as shown below) and populates the Layer 2 forwarding database (FDB) on the SPB bridges with MAC addresses.

Bridge B
MAC: B, priority: 2

Bridge B
MAC: B, priority: 2

B, C

ROOT, Bridge A **D**
MAC: A, priority: 1

Bridge C
MAC: C, priority: 3

Bridge A
MAC: A, priority: 1

Bridge C
MAC: C, priority: 3

A, B

C

Bridge D
MAC: D, priority: 4

ROOT, Bridge D
MAC: D, priority: 4

(1)

(2)

ROOT, Bridge B
MAC: B, priority: 2

Bridge B
MAC: B, priority: 2

A, D

C

Bridge A
MAC: A, priority: 1

Bridge C
MAC: C, priority: 3

Bridge A
MAC: A, priority: 1

A, B

D **ROOT**, Bridge C
MAC: C, priority: 3

Bridge D
MAC: D, priority: 4

Bridge D
MAC: D, priority: 4

(3)

(4)

Figure 3: ISIS-SPB Shortest Path Calculations

The ISIS-SPB unicast trees shown in Figure 3 were built as follows:

1 Bridge A calculates the shortest path tree to Bridge B and then programs its FDB with MAC address B on the link, as shown in (1).

2 Bridge A will then calculate shortest paths to Bridge C and Bridge D and programs the MAC addresses according to the path computed.

3 All other bridges follow the same procedure (note that the actual computation is much more optimized and the description here is only for illustration purposes).

4 The following traffic pattern for this example network is the result of the ISIS-SPB SPT calculations:
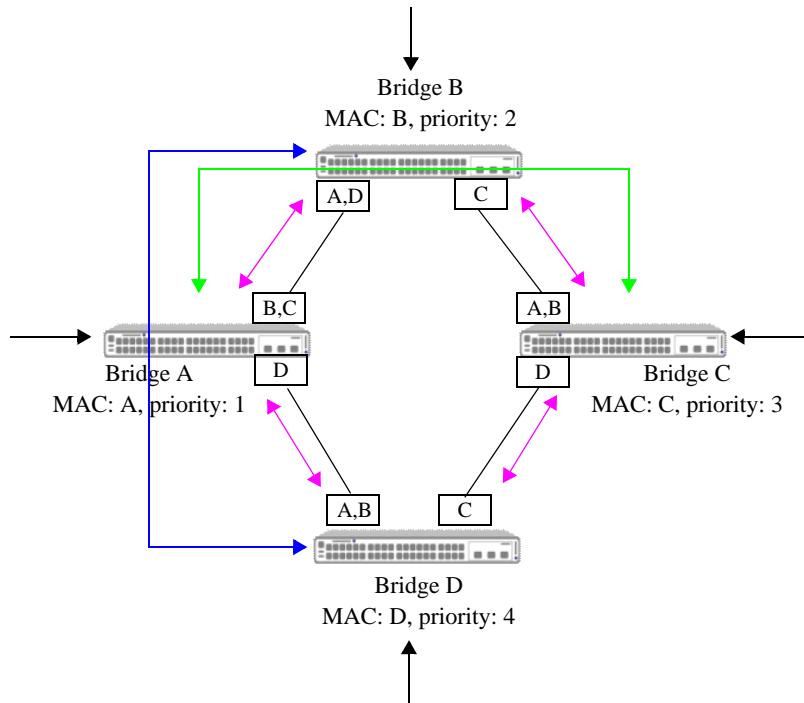
Figure 4: ISIS-SPB Topology

As shown in Figure 4, all the backbone MAC (BMAC) addresses are learned by the switches when ISIS-SPB converges. The path taken by each unicast flow (for example ABC, CBA) are reverse path congruent and travel the shortest path through the network.

In the ISIS-SPB topology (Figure 4), the link between Bridge D and Bridge C carries traffic, whereas in the Spanning Tree topology (Figure 3), this link is blocked. Although these examples are based on traffic distribution for a single BVLAN, the ability to make all links in the topology available at all times is especially advantageous in highly redundant, meshed networks.

Although the link between Bridge D and Bridge C is used in the ISIS-SPB topology, traffic flow is relatively low in comparison to the other links. To make better use of this link, a second BVLAN could be created and assigned a different ECT algorithm to trigger ISIS-SPB calculations of a separate set of SPTs for the second BVLAN. This is similar to creating a new Multiple Spanning Tree (MST) instance in a Spanning Tree topology to create a different tree and assigning a new VLAN to that instance.

Each ECT algorithm uses a different calculation to break ties when paths between SPB bridges are equal cost. Another method to influence the SPT calculation is to modify the bridge priority for the switch or change the link cost metric for the SPB interface connection between two switches.

Multicast Traffic

SPBM supports two methods for replicating and forwarding multicast traffic (or unknown destination traffic) received from customer equipment: head-end replication and tandem replication.

- **Head-end replication**. Multicast traffic is replicated once for each receiver, encapsulated with the BMAC address, and then sent as a unicast packet to each destination. This method is more suited for networks where there is a low demand for multicast traffic.

- **Tandem replication**. Multicast traffic is replicated only where there is a fork in the SPT and each branch has at least one receiver. Each multicast source bridge in the SPBM network is the root for a

multicast distribution tree (MDT). An MDT is created per-source per-BVLAN and it is pruned according to whether the SPB node is on the shortest path of a multicast transmitter and receiver. For those MDTs that cross a given Backbone Core Bridge (BCB), that BCB needs to generate a multicast forwarding table for each such MDT.

Multicast traffic originating from a bridge is encapsulated with a special multicast BMAC DA that identifies the source of the traffic and then forwarded on the tree. Participating bridges that receive the packet will then know the source of the traffic and will use the multicast forwarding information for that source to switch the packet to the appropriate destination.

SPB Services

The SPBM network topology consists of two layers:

- **The backbone infrastructure (control plane) layer**. ISIS-SPB builds the backbone layer by defining loop-free, shortest path trees (SPTs) through the backbone network (see "SPBM Shortest Path Trees" on page 3-7 for more information).

- **The services (data plane) layer**. The service layer is based on the Provider Backbone Bridging (PBB) framework as defined in the IEEE 802.1ah standard. SPBM supports the 802.1ah MAC-in-MAC method for data encapsulation. SPBM services transport the encapsulated traffic over the ISIS-SPB infrastructure.

The SPB service layer framework is comprised of the following components:

- **Backbone Edge Bridge (BEB)**. An OmniSwitch is considered a BEB if the switch is SPB capable and at least one service access point (SAP) and one SPB interface is configured on the switch. The BEB marks the boundary between the customer network and the PBB network (PBBN).

- **Backbone Core Bridge (BCB)**. An OmniSwitch is considered a BCB if the switch is SPB capable and no SAPs are configured but at least one SPB interface is configured on the switch to forward encapsulated SPBM network traffic. Note that the requirement for configuring a BCB is based on whether or not the network topology includes a transit bridge.

- **Service Instance Identifier (I-SID)**. Configured only on a BEB, this component identifies a backbone service instance that will tunnel the encapsulated data traffic through the PBBN between BEBs. The I-SID is bound to a BVLAN ID and a Service Manager SPB service ID when the service is created.

- **Access Port**. A port or link aggregate configured as an SPB access port. This type of port is configured on the BEBs and defines the point at which traffic from other provider networks or directly from customer networks enters the PBBN. The access port is also associated with a Layer 2 profile that specifies how to process protocol control frames received on the port

- **Service Access Point (SAP)**—A SAP is a logical service entity (also referred to as a virtual port) that is configured on a BEB to bind an access port to an SPB service ID and specify the type of customer traffic ((untagged, single-tagged, double-tagged, or all) to encapsulate and tunnel through the PBBN.

- **SPB Interface (Network Port)**—A port or link aggregate configured as an SPB interface that resides on either a BEB or a BCB and connects to the backbone network. Network ports carry customer traffic encapsulated in 802.1ah frames and are associated with all BVLANs on the switch. Customer traffic ingressing on a network port is switched to another network port (on BCBs) or to an access port (on BEBs).

Once the ISIS-SPB infrastructure and the SPB service-based architecture is defined, the following service components are dynamically created by the OmniSwitch. No user-configuration is required.

- **Service Distribution Point (SDP)**—A SDP provides a logical point at which customer traffic is directed from one BEB to another BEB. SDPs are used to set up distributed services, which consist of at least one SAP on a local node, one SAP on a remote node, and an SDP binding the service on both nodes.

- **SDP Bind**—An SDP binding represents the binding of an SPB service instance to an SDP. The SDP then distributes the service connectivity to other BEBs through the ISIS-SPB shortest path trees.

Sample SPBM Network Topology

The following diagram provides a sample SPBM network topology that shows how the SPBM service and ISIS-SPB backbone layers work together to basically extend (or virtualize) customer traffic across a Provider Backbone Bridge Network (PBBN):
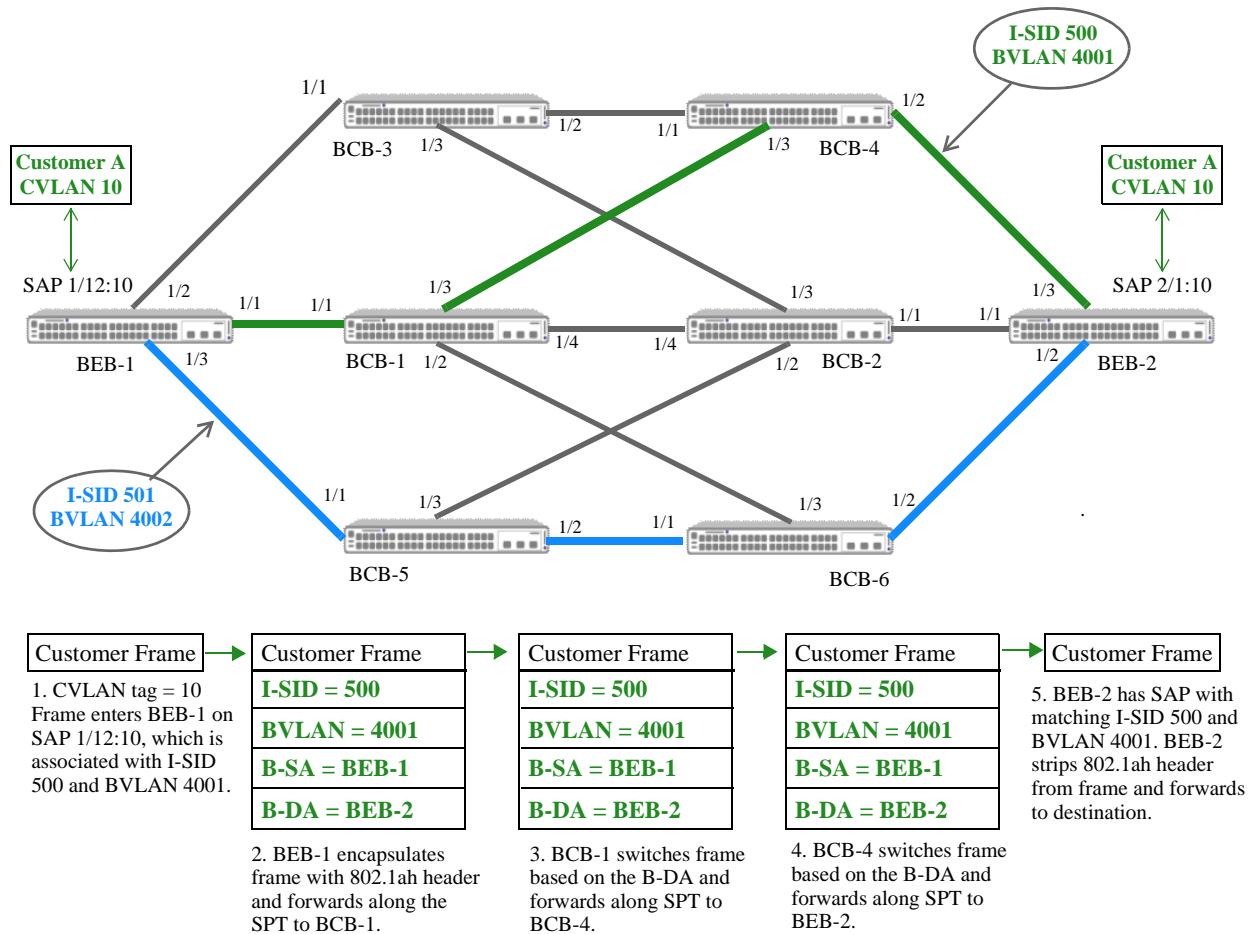
Figure 5: Sample SPBM Network

In this sample SPBM topology:

- The packet flow for Customer A frames tagged with VLAN 10 is shown as a typical example. These frames are mapped to an SPB service that represents a binding of I-SID 500 to BVLAN 4001. The path for this binding is shown in green.

- An additional path, shown in blue, is for another SPB service that represents a binding of I-SID 501 to BVLAN 4002. This provides an example of how adding an additional BVLAN and service

configuration to the topology can provide an alternate service path for other traffic from the same customer or traffic from a completely different customer.

- SPB BVLAN 4001 and 4002 are created and assigned to ECT ID 1 and 2, respectively, on every switch (BEBs and BCBs) in the topology. These BVLANs serve as the transport entity on which ISIS-SPB builds the shortest path trees and SPB services tunnel data.

- The switch ports connecting each SPB switch with the next-hop SPB switch are configured as SPB interface ports. This type of port is used to forward ISIS-SPB control packets and serves as a network port for tunneling encapsulated traffic through SPB services.

- The service access points (SAPs) created on BEB-1 and BEB-2 determine which frames from Customer A are accepted on the SAP port, where they are then encapsulated and mapped to the associated service. Other SAPs exist on these switches for the other service path.

- When a frame tagged with VLAN 10 ingresses on port 1/12, the frame is encapsulated in an 802.1ah header. The header specifies the BMAC for BEB-1 as the B-SA, the BMAC for BEB-2 as the B-DA, the SAP I-SID (500), and the SAP BVLAN (4001).

- All other frames ingressing on SAP 1/12:10 that are not tagged with VLAN 10 are dropped, unless there are other SAPs configured for that port that will classify those frames.

- The encapsulated frame is then forwarded along the BVLAN 4001 shortest path tree (SPT) to BEB-2, where the 802.1ah header is stripped off and the frame is forwarded to the appropriate destination port.

- The entire process for encapsulating and tunneling customer frames is the same for frames ingressing on port 2/1 of BEB-2 destined for BEB-1.

How it Works

- There is one instance of ISIS-SPB supported in the backbone topology. This instance is activated once the BVLANs and SPB interfaces are created and the administrative status of ISIS-SPB is enabled for each switch.

- When ISIS-SPB is administratively enabled on each switch, all the configured SPB interfaces start to advertise Hello packets to discover and establish adjacencies with other SPB switches.

- Once adjacencies are established, link state packets (LSPs) are generated with SPB-specific TLVs and shortest path trees from each switch to all other switches are calculated.

- Each SPB switch learns the backbone MAC (BMAC) address and associated BVLAN IDs of every SPB switch in the network and stores that information in a local forwarding database. The BMAC address is the bridge MAC address of the switch and is advertised by ISIS-SPB as the System ID.

- ISIS-SPB then informs Service Manager of the reachability of the BMAC/BVLAN combinations. This information is used to automatically create a service distribution point (SDP) between the same BVLAN on each BEB.

- When ISIS-SPB receives advertisement of a service instance identifier (I-SID) from a remote BEB that matches an I-SID created on the local switch, the SDP (BMAC/BVLAN) of the remote BEB is bound to the I-SID. The binding of a service to an SDP is referred to as a mesh SDP.

- Basically, an SDP is a dynamically created logical entity that distributes service connectivity to other BEBs through the ISIS-SPB shortest path trees.When customer frames are then classified into a specific SAP, the frames are encapsulated and tunneled through the mesh SDP (service/SDP bind) associated with that SAP.

Remote Fault Propagation for SPBM Services

When a point-to-point connection is emulated with a Layer 2 SPB service, it is necessary to propagate connectivity faults from one end of the service tunnel to the other end. This allows a locally connected device to detect a connectivity fault in the SPB service and take action (such as enable a redundant link or send a trap) in response to the detected fault. Remote Fault Propagation (RFP) for SPB provides this type of fault detection and propagation from one end of an SPB service to the other.

The RFP functionality is applied to the SPBM service (data plane) layer. Connectivity fault events are propagated into an SPB Service Access Point (SAP). A SAP is associated with an SPB access port and a service instance identifier (I-SID). When a SAP port goes down, the SAP port on the other end of the service is also brought down. Without the RFP for SPB feature, the other end would continue to transmit packets waiting for a response.

Ethernet OAM messaging is used to detect a failed condition and propagate the fault. An OAM Continuity Check Message (CCM) is sent at specified intervals between SAPs to advertise the status of SAP components (such as the SPB access port and I-SID information).

This implementation of RFP for SPB involves setting up the following components:

- An underlying SPB network infrastructure. RFP will monitor SPB access ports, which are bound to SAPs. A SAP consists of an access port, SPB service ID, and an encapsulation value (the VLAN tags that the SAP will process on the access ports).

- An RFP domain, which consists of local maintenance end points (MEPs) with remote end point lists that are assigned to the same RFP domain ID.

  - A local MEP defines the RFP domain parameters, such as the RFP domain ID, level, and CCM interval. An ID number is assigned to the local MEP to identify the local switch as a participant in an RFP OAM domain.

  - A remote end point list identifies the SPB services to monitor and the remote end points (the MEP IDs of remote switches) to which the status of the services is advertised. Configuring the remote end point list of an RFP domain triggers the sending of CCM packets.

- A reserved Ethernet OAM domain to which the RFP domain is mapped. When the local MEP of an RFP domain is configured, an OAM domain is automatically created based on the parameters specified when the local RFP MEP was created.

The following diagram shows how RFP works in a sample SPBM network topology:
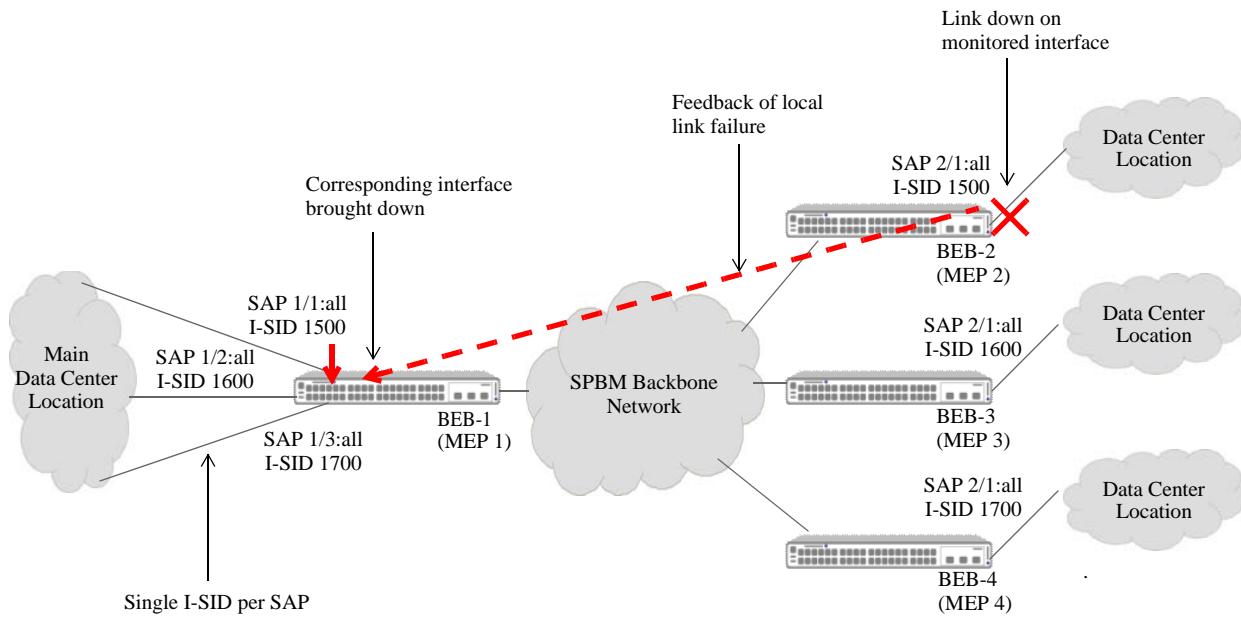
Figure 6: RFP in a Sample SPBM Network

In this sample SPBM topology:

- An RFP local MEP and a remote end point list are configured on each Backbone Edge Bridge (BEB). Both are assigned to the same RFP domain ID on each BEB to identify the end points as participating members of the RFP domain.

- Each local MEP is assigned an ID number, which is used as the virtual UP MEP ID. In this example, the virtual UP MEP ID is 1 for BEB-1, 2 for BEB-2, 3 for BEB-3, and 4 for BEB-4.

- Each remote end point list specifies the SPB services to monitor and the MEP IDs of remote BEBs to which the status of the services is advertised. For example, the remote end point list on BEB-1 contains the monitored SPB services and local MEP IDs for BEB-2, BEB-3, and BEB-4.

- The remote end point list binds an SPB service ID to the RFP domain. The service ID is associated with a service instance identifier (I-SID) and a SAP, which identifies the SPB service instance and access port to monitor. For example, on the BEB-2 switch, the status of I-SID 1500 on access port 2/1 is monitored and advertised to BEB-1.

- CCM packets transmitted on the RFP domain advertise I-SID and access port status information for the local SAP. The SAP information to advertise is identified through the SPB service ID that is associated with the RFP domain. For example, BEB-1 and BEB-2 both advertise the status of SAP port 1/1 and SAP port 2/1 for I-SID 1500. The same SPB service ID is mapped to each of these SAPs, which means the same I-SID is mapped to each of these SAPs.

- When port 2/1 goes down on BEB-2, the service represented by I-SID 1500 stops transmitting. The CCM packets transmitted between BEB-2 and BEB-1 detect and advertise the port down fault. This causes BEB-1 to administratively down port 1/1 in response to the fault condition.

For an example of the CLI configuration for this sample deployment of RFP in an SPB network, refer to the.

Customized CCM Packets

The CCM packets transmitted between RFP end points contain a proprietary OUI TLV that provides link fault information for the SPB services that are monitored by the RFP domain. The following shows an example of the proprietary TLV format:

Type = 127	Length	OUI MAC == Alcatel OUI (3 octet) Information about ISID, Portstate ISID 24 bit value (3 octet) Port state (UP/STATE) (1 octet) The above information is repeated if there are multiple I-SIDs.

- OAM unicast CCM packets are sent without Provider Backbone Bridge (PBB) header encapsulation across the SPBM network to each remote BEB device on the control B-VLAN.

- Only information related to the I-SID associated with the remote BEB is sent in the proprietary TLV.

- OAM packets are filtered on the SPB SDP interfaces to capture only the CCM packets used for RFP monitoring.

- Only CCM information will be processed for the related I-SID information on the receiving switch.

Fault Detection

Each BEB in the RFP domain will check the I-SID and port state information contained in the received CCM packets.

- If any port state has transitioned from up to down, the local SAP port associated with the same I-SID is also brought down as a port violation.

- When a CCM indicates that the downed port has transitioned back to an up state, the local port violation is cleared.

- After a port violation is cleared, a 10 second timer is started to avoid bringing down the local ports immediately. This allows for the scenario in which a port violation is manually cleared on one BEB and by the time the violation is cleared on another BEB, a CCM packet from the other BEB is received with SAP port down information.

- If a BEB device goes down, the information about the BEB will time out on remote BEB devices after 3 multiplied by the value of the CCM interval (3*CCM interval value). For example, if the CCM interval value is set to 100ms, a remote BEB will wait 300ms before timing out the information about the BEB that went down. The local physical SAP access port mapped to the I-SID that timed out is then brought down as well.

For more information and CLI configuration examples, see “Configuring Remote Fault Propagation for SPBM” on page 3-49 and “RFP for SPB Configuration Example” on page 3-53.

For more information about Ethernet OAM, see Chapter 37, “Configuring Ethernet OAM.”

IP over SPBM

The OmniSwitch implementation of SPBM provides L2 VPN capability that bridges L2 customer LAN segments. Customer edge (CE) devices form peers and exchange routing information, as well as perform the necessary IP forwarding. Then the SPBM BEBs bridge the already routed IP traffic across the SPBM backbone.

In addition to L2 VPN, the OmniSwitch also provides an IP over SPBM capability that consolidates the routing functionality of CE devices into the BEB devices. The Virtual Routing and Forwarding (VRF) instances on different BEBs are tied together via backbone I-SIDs across the same SPBM backbone that is used to support Layer 2 VPNs.

The OmniSwitch IP over SPBM solution supports two methods for combining L3 routing and L2 SPBM in the same switch: VPN-Lite and L3 VPN.

VPN-Lite

The VPN-Lite method provides a gateway between a regular SPBM service and a router within the same OmniSwitch chassis. This solution provides a specific advantage in that it allows a single box to represent two tiers in a typical fat-tree network, which is popular in data center deployments.

In addition, a VPN-Lite configuration can act purely as an L3 VPN when configured correctly. In this mode, existing routing protocols can form adjacencies across the SPBM PBB network. To keep it purely an L3 VPN, the administrator makes sure that no SPBM SAPs that can inject bridged flows are allowed to attach to the I-SID designated for the specific VPN.

The VPN-Lite approach uses the SPBM network in the same way a VLAN is used for transporting L3 frames. Each BEB or host can inject frames into the I-SID as needed, and BEBs can decide to bridge or route those frames based on their inner and outer destination MAC address.

L3 VPN

When the L3 VPN method is implemented, the OmniSwitch acts as an access or edge router to multiple VRFs and connects these VRFs across an SPBM PBB network. Each VPN is identified by a local VRF instance on each BEB and globally in the backbone by an I-SID in the PBB header. ISIS-SPB will import and export routes from the local routing protocols running inside their respective VRFs. In essence, ISIS-SPB is creating tunnels between BEBs through which routed frames are sent to reach their target networks.

The OmniSwitch L3 VPN solution is based on the IETF drafts *IP/IPVPN services with IEEE 802.1aq SPB(B) networks* and uses IS-IS TLVs to exchange routes between the BEBs that host the same VPN services. This approach also gives an administrator the ability to build VPNs and extend them over an SPBM core.

L3 VPN Interface

An L3 VPN interface serves as an IP gateway to access remote networks and is required when using either the VPN-Lite or L3 VPN method. There are two options for defining an L3 VPN interface:

● Configuring a service-based IP interface (OmniSwitch 9900 only). When this option is used, an IP interface is created and bound to an SPB service ID. The service ID is associated with an I-SID.

● Configuring an external loopback port configuration (all supported OmniSwitch platforms, except the OmniSwitch 9900). When this option is used, a physical cable connects a regular port to a service access port. The regular port is tagged with an IP interface VLAN; the access port is associated with an SPB Service Access Point (SAP). The VLAN-based IP interface serves as the L3 VPN interface.

The external loopback configuration option defines an L3 VPN loopback interface, and the service-based IP interface option defines an L3 VPN service-based interface.

L3 VPN Service-Based Interface

Configuring an L3 VPN service-based interface is an option that is only available on an OmniSwitch 9900 (no other OmniSwitch platforms support this option). Using the service-based interface option simplifies the configuration in that a physical cable and two switch ports are not required to form an external loopback. Instead, an IP interface is created within a VRF instance and bound to an SPB service. This connects routes within the VRF instance to an SPB service instance (I-SID).

The following diagram shows a logical depiction of an L3 VPN service-based interface configuration on an OmniSwitch 9900 (no physical cable is required):
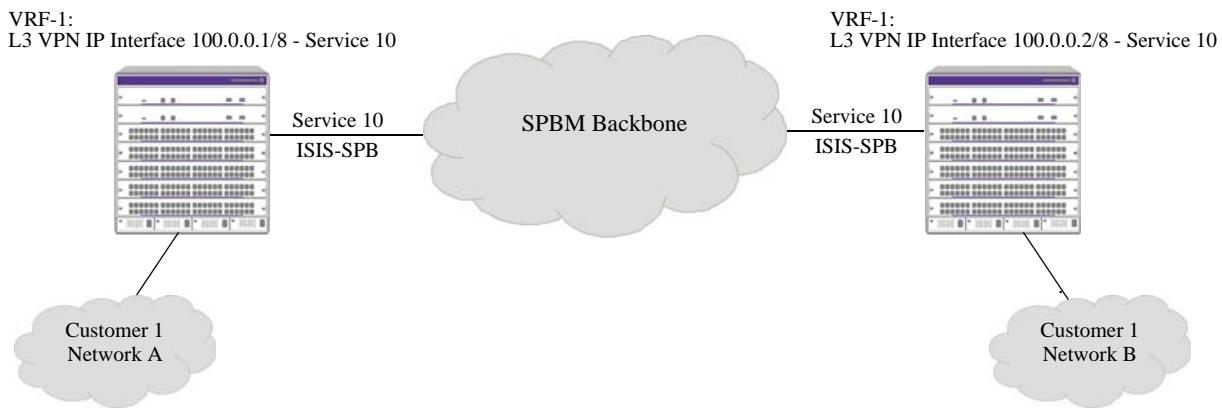
Figure 7: L3 VPN Interface (No Loopback)

L3 VPN Loopback Interface

Configuring a physical loopback port configuration to define an L3 VPN interface is an option that is only available on the OmniSwitch 6900, OmniSwitch 6860, and OmniSwitch 6865 (the OmniSwitch 9900 does *not* support this option). A regular switch port or a static link aggregate can serve as a loopback port. In addition, multiple loopback port pairs are allowed and can be shared between different VRFs.

An L3 VPN loopback interface configuration consists of the following components:

- An IP interface created in a specific VRF instance and bound to a VLAN ID.

- A regular switch port or link aggregate tagged with the IP interface VLAN.

- A service access port that is assigned to an SPB SAP. The SAP encapsulation is configured with the VLAN ID that is tagged on the regular switch port or link aggregate.

- A physical cable that connects the VLAN port with the SAP port to form the loopback configuration.

The loopback configuration connects a VRF to an SPB SAP and can carry traffic from different VRFs tagged with different VLANs. The following diagram shows a logical depiction of an L3 VPN loopback interface:
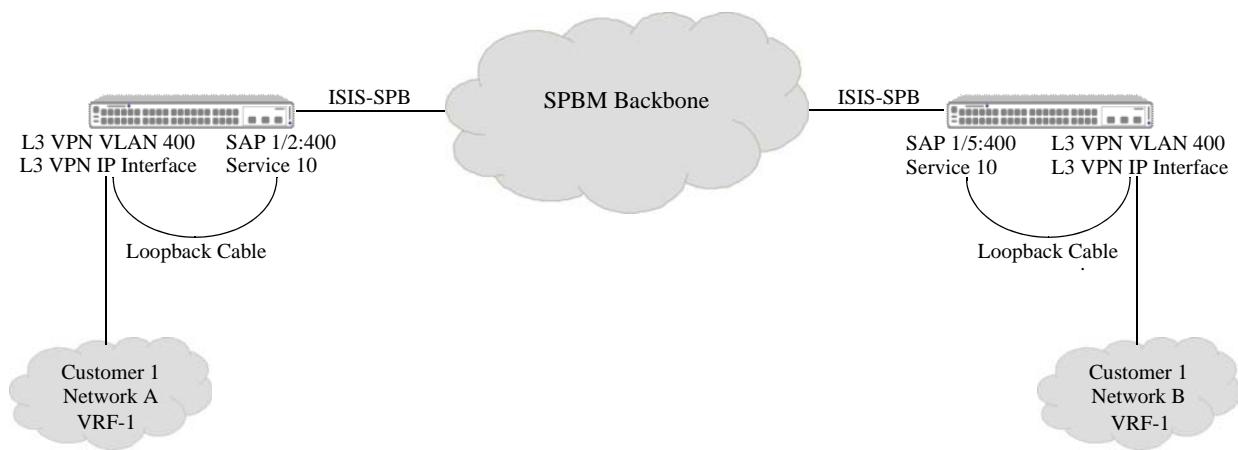
Figure 8: L3 VPN Loopback Interface

How it Works

This section describes the VPN-Lite and L3 VPN control and data plane operations in an IP over SPB network configuration. Although both approaches use an L3 VPN interface configuration, they differ in how routing protocol control packets are exchanged and processed to support IP over SPB.

VPN-Lite Control Plane Operations

When routing protocols or static routes are running on the L3 VPN interface, the interface can exchange IP routes with other L3 VPN interfaces that are running the same routing protocols and are associated with the same I-SID. By exchanging routes with other L3 VPN interfaces, VRFs on different BEBs can learn remote networks from each other.

- If an OmniSwitch 9900 is used in this scenario without a loopback port configuration, the routing protocol control packets are sent from the L3 VPN service-based IP interface and carried on SDPs into the SPBM backbone. The control packets received from the SPBM backbone travel from SDPs to the VRF following the same process but in reverse.

- If a physical loopback port configuration is used in this scenario, the routing protocol control packets sent from the L3 VPN IP interfaces travel though the tagged VLAN port, enter the service access port, where the packets are then distributed into different services by SAPs associated with the access port and carried on SDPs into the SPBM backbone. The control packets received from the SPBM backbone travel from SDPs to the VRF following the same process but in reverse.

L3 VPN Control Plane Operations

The ISIS-SPB support of the IPVPN TLV and IPv4 sub-TLV provides a different method for exchanging L3 routes between VRFs. Instead of running routing protocols on the L3 VPN IP interfaces, IP routes are imported into ISIS-SPB from VRFs. ISIS-SPB then carries these routes in the TLVs through the SPBM cloud to other SPBM BEBs. When ISIS-SPB receives IPVPN TLVs from the cloud, ISIS-SPB will export the routes to the appropriate VRFs.

The L3 VPN approach implements the importing and exporting of routes between ISIS-SPB and VRF instances and the transport of these routes using the supported TLVs. The administrator does not configure routing protocols on the L3 VPN IP interface. Implementing the L3 VPN approach requires careful consideration to avoid routing loops.

VPN-Lite and L3 VPN Data Plane Operations

Data is moved in the same manner for both VPN-Lite and L3 VPN traffic, and the existing data plane forwarding mechanisms for SPB and IP are used without modification:

- An L3 VPN interface serves as an IP gateway to access remote networks. The network administrator has to ensure the IP subnet reachability of the L3 VPN addresses on the same SPBM I-SID.

- L3 VPN interfaces use dynamic ARP to learn the MAC addresses of other L3 VPN interfaces and provide next-hop forwarding information to the switch.

- IP data plane packets travel the same path as VPN-Lite control packets (see "VPN-Lite Control Plane Operations" on page 3-19 for more information).

- Data in the SPBM cloud is encapsulated into the Provider Backbone Bridge (PBB) format (see "SPB Services" on page 3-11 for more information).

For more information and configuration examples, see "Configuring IP over SPB" on page 3-56 and "IP over SPB Configuration Examples" on page 3-58.

Interaction With Other Features

This section contains important information about Shortest Path Bridging MAC (SPBM) interaction with other OmniSwitch features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

Backbone VLANs (VLAN Manager)

VLAN Manager CLI commands are used to create an SPB backbone VLAN (BVLAN). Although a BVLAN is created in a similar manner as a standard VLAN, BVLANs differ from standard VLANs as follows:

- No Spanning Tree control—the Spanning Tree protocol is automatically disabled on each BVLAN and all ports associated with each BVLAN will remain in a forwarding state. However, Spanning Tree can remain operational on other types of VLANs.

- No source MAC address learning—normal hardware learning is disabled on BVLANs. Instead, the forwarding database (FDB) is populated by the ISIS-SPB protocol.

- There is no flooding of unknown destination or multicast frames.

- Ingress filtering based on the source MAC address—frames received on ports that do not have an incoming source MAC address pre-programmed by ISIS-SPB are discarded.

- IP interfaces are not supported on BVLANs.

IP Multicast Switching

In a networking environment where IP multicast traffic is used, destination hosts signal their intent to receive a specific IP multicast stream by sending an Internet Group Management Protocol (IGMP) request to a nearby switch. This process is referred to as IGMP Snooping. The switch then learns on which ports multicast group subscribers are attached and can intelligently deliver traffic only to the respective ports. The OmniSwitch implementation of IGMP Snooping is called IP Multicast Switching (IPMS).

IGMP Snooping for SPB services is essentially the same. An SPB Backbone Edge Bridge (BEB) will apply the logic of IGMP Snooping on a per-service basis to limit the traffic going out of each Service Access Point (SAP) port, as well as limit traffic going out across each backbone port. The SPB bridge will monitor the IGMP queries and requests from SAPs and Service Distribution Point (SDP) ports (also referred to as network virtual ports) to build the stream membership association logic and timing in the same manner as is done on a standard IGMP Snooping bridge.

When traffic arrives on a SAP port, the switch will examine the packet to see if there are any known receivers. If there are any such receivers, then only ports (including network virtual ports) will have a copy of that frame sent on them. When traffic arrives from the core on a network virtual port, the same logic is applied so that a copy of the frame is only sent out on a port where a listener has requested membership to the stream. However, traffic from the core is never sent back into the core (split horizon protection).

IPMS is configurable in both the VLAN and service domains. Enabling IPMS functionality specifically for SPB services is required to activate IGMP Snooping in an SPB network. See the "Configuring IP Multicast Switching" chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide* for more information.

Link Aggregation

- Both static and dynamic link aggregates are configurable as SPBM service access ports and as SPBM network interfaces.

- Note that a link aggregate must consist of all access ports or all network ports. SPBM functionality is not supported on link aggregates that consist of a mixture of SPBM ports and standard switch ports.

- When creating a link aggregate that will serve as an SPBM service access port or network interface, specify the Tunnel Protocol hashing option for the aggregate. This will ensure that hashing is done on the payload of encapsulated SPB packets. See the "Link Aggregation Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information.

OAM

- OAM support per the IEEE 802.1ah standard for Provider Backbone Bridging (PBB) is applied at the customer VLAN (CVLAN) and the backbone VLAN (BVLAN) level. Support at the service instance (I-SID) level is provided through Remote Fault Propagation on SPB.

- The OmniSwitch Ethernet OAM feature is required to support RFP for SPB. When an RFP local end point is created on the switch, the following reserved maintenance domain and maintenance association is created:

  - RFP_OVER_SPB_DOMAIN_LEVEL*x* (where *x* is the level number specified when a local RFP end point is created)
  - RFP_OVER_SPB_ASSOCIATION

- In addition, the OmniSwitch proprietary Layer 2 ping and traceroute features are available to troubleshoot CVLAN and BVLAN domains, including an I-SID check.

Quality of Service (QoS)

- The priority assignment of a user frame is determined at an access point. A Service Access Point (SAP) on an SPB access port can be configured as trusted or un-trusted. If a SAP is configured as trusted, then internal priority for ingress traffic on that SAP is derived from tagged or NULL tagged ingress packet priority or from default port priority if ingress packet is untagged. If a SAP is untrusted then internal priority can be configured by the user.

- QoS performs the following actions on ports configured as access ports:

  - Access ports are automatically trusted and the default classification is set to 802.1p.
  - The trust status and classification are not user-configurable on access ports.
  - All QoS CLI configuration is blocked on access ports.This includes physical ports and ports that are members of a link aggregate.
  - Untagged L2 control packets (such as BPDU, GVRP, AMAP) are always tunneled (if enabled) through the SPB domain with the default EXP bits set at 7, so that they can arrive at the destination at the highest priority of 7. Trusted/untrusted SAPs configured on access ports will not affect the priority assignment for Layer 2 control packets.

- QoS priority (802.1p) is applied as follows to trusted and untrusted SAPs:

SAP Configuration	Allowed Configuration	
Tagged (VLAN 1–4094)	**Trusted**	**Tagged traffic priority derived from tags.**
	Untrusted	**Tagged traffic priority configured by user.**

SAP Configuration	Allowed Configuration	
QinQ (outer VLAN 1–4094)	Trusted	Tagged traffic priority derived from outer tags.
	Untrusted	Tagged traffic priority configured by user.
Wild Card	Trusted	Tagged traffic priority derived from tags. Untagged traffic Port default (PRI 0).
	Untrusted	Tagged/ traffic priority configured by user
Untagged	Trusted	Untagged Traffic Port default (PRI 0)
	Untrusted	Priority configured by user.

- – By default, a SAP is trusted with best effort priority (0).
- – A SAP can be dynamically changed to trusted/untrusted without administratively taking down the SAP.
- – A SAP priority may only be set when a SAP is untrusted.
- – When a SAP is changed from untrusted to trusted, any previously assigned priority is reset with best effort priority (0).
- – A trusted SAP that defines a double-tagged encapsulation (QinQ) will use the outer VLAN tag to determine the priority of the frame.

- Priority handling at the edge and core components of an SPBM topology:

- – On a ingress Backbone Edge Bridge (BEB), a frame is classified to a SAP. The internal priority is determined based on the QoS settings of the SAP (for example, trusted vs. untrusted, default priority). This internal priority is mapped to the backbone VLAN (BVLAN) tag of the tunnel encapsulation.
- – The Backbone Core Bridge (BCB) acts as a Layer 2 device that switches the frame across ingress to egress ports in the BVLAN domain. The BVLAN tag is used to determine the internal priority queue on the egress port where the frame is enqueued.
- – On an egress BEB, the internal priority is determined from the BVLAN tag. The frame is de-encapsulated and enqueued to the egress queue(s) of the access port(s) based on this internal priority.

Universal Network Profiles (UNP)

Integration with Virtual Machine Network Profiles (vNPs) to support device discovery and mobility. The UNP feature supports two types of profiles: VLAN and service. A service profile can be configured to classify traffic for SPB or VXLAN tunneling.

The OmniSwitch supports both a VLAN and service domain for traffic classification.

- The VLAN domain is identified by a VLAN ID. In the VLAN domain, each VLAN is accessed through a physical port. Each physical port can have more than one VLAN attached. UNP VLAN classification associates a MAC address to a specific VLAN on a physical UNP bridge port.

- The service domain is identified by one of the following:

- – A Shortest Path Bridging (SPB) service instance identifier (I-SID), which is associated with a Service Manger service ID to represent a virtual forwarding instance (VFI).
- – A VXLAN Network Identifier (VNI), which is associated with a Service Manager service ID to represent a VFI.

In the service domain, each VFI is accessed through a virtual port, referred to as a Service Access Point (SAP). UNP service classification associates a device MAC address to a SAP.

Dynamic Service Access Points

A UNP service profile can trigger the dynamic creation of a SAP when traffic received on a UNP access port is classified and assigned to that profile. If the service (SPB or VXLAN) that the SAP is associated with does not exist, the service is also dynamically created.

Allowing incoming traffic to trigger dynamic SAP creation reduces the amount of manual configuration required. In addition, no other protocols are required on the switch or host device to support this functionality.

UniDirectional Link Detection (UDLD)

UDLD protocol control frames (destination MAC address is 01:00:0c:cc:cc:cc) are processed as follows:

UDLD Status	User Access Port	Network Port (Tagged)	Network Port (Untagged)	Legacy
Globally disabled	tunnel	tunnel	discard	tunnel
Globally enabled	tunnel	tunnel	discard	drop
Enabled on port	peer	tunnel	peer	peer

VRF

IP over SPB uses Virtual Routing Forwarding (VRF) instances to exchange routes with I-SIDs. This is accomplished via the Global Route Manager (GRM). VRF routes are exported to the GRM table and imported into I-SIDs; I-SID routes are exported to the GRM table and imported into VRFs.

- A binding is created between a VRF and the I-SIDs to identify which I-SIDs will export routes to the GRM for the specified VRF (see "Configuring IP over SPB" on page 3-56 for more information).

- The **ip import** command has an optional **isid** parameter that notifies GRM to import the routes from the specified I-SID into the requesting VRF.

Quick Steps for Configuring SPBM

This section provides a quick tutorial for configuring the SPBM network backbone (control plane) and the service encapsulation path (data plane). The Command Line Interface (CLI) commands provided in this section are used to configure the "Sample SPBM Network Topology" on page 3-12.

Quick Steps for Configuring the SPBM Backbone

The following quick steps are used on each switch in the SPBM backbone that will participate in the "Sample SPBM Network Topology" on page 3-12. This includes both edge and transit (core) switches.

1 Use the **system name** command to assign a unique system name to each SPB switch in the domain.

```
-> system name BEB-1
-> system name BEB-2
-> system name BCB-1
-> system name BCB-2
-> system name BCB-3
-> system name BCB-4
-> system name BCB-5
-> system name BCB-6
```

2 Use the **spb bvlan** command to create BVLANs 4001 and 4002 on each switch (edge and core switches) that will participate in the SPBM topology.

```
-> spb bvlan 4001
-> spb bvlan 4002
```

3 Use the **spb isis bvlan ect-id** command to change the equal cost tree (ECT) algorithm ID for the specified BVLAN, if necessary, to make sure that the same ECT ID is assigned to the same BVLAN ID on each switch (edge and core switches) in the SPBM topology.

```
-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2
```

4 Use the **spb isis control-bvlan** command to designate one of the BVLANs on each SPB switch (edge and core switches) as the control BVLAN for the SPB instance. The control BVLAN is used to exchange ISIS-SPB control packets with neighboring SPB switches.

```
-> spb isis control-bvlan 4001
```

5 Use the **spb isis interface** command to configure a port or link aggregate as an SPB interface. This type of interface sends PDUs to detect neighboring SPB switches and form adjacencies and also serves as a network port that is used to carry encapsulated service traffic through the SPBM backbone network.

```
-> spb isis interface port 1/1-4
-> spb isis interface port 1/1-3
```

In the "Sample SPBM Network Topology" on page 3-12, ports 1/1-4 are configured as SPB interfaces on the core switches (BCB-1, BCB-2) and ports 1/1-3 are configured as SPB interfaces on all other switches.

6 Use the **spb isis admin-state** command to enable the SPB instance for the switch. Enabling ISIS-SPB on the switch triggers the transmission of hello packets from the SPB interfaces, which starts the process of defining the SPB infrastructure and calculating the shortest path trees (SPTs) through the topology.

```
-> spb isis admin-state enable
```

Quick Steps for Configuring SPB Services

The following quick steps use the OmniSwitch Service Manager commands to configure the logical entities that comprise the SPB services in the "Sample SPBM Network Topology" on page 3-12.

1 Use the **service access** command to configure a port or link aggregate on which customer traffic is received as an SPB service access port.

```
-> service access port 1/12
-> service access port 2/1
```

2 Use the **service spb** command to create an SPB service and associate that service with a backbone service instance identifier (I-SID) and BVLAN.

```
-> service 1 spb isid 500 bvlan 4001 admin-state enable
-> service 2 spb isid 501 bvlan 4002 admin-state enable
```

3 Use the **service sap** command to create a service access point (SAP) by associating an SPB service with SAP ID. A SAP ID is comprised of a port or link aggregate and an encapsulation value that identifies the customer traffic to associate with the service.

```
-> service 1 sap port 1/12:10 admin-state enable
-> service 1 sap port 2/1:10 admin-state enable
-> service 2 sap port 1/12:0 admin-state enable
-> service 2 sap port 2/1:all admin-state enable
```

In this example,

- SPB service 1 (I-SID= 500, BVLAN=4001) is assigned to SAPs 1/12:10 and 2/1:10. Traffic received on the SAP access ports (1/12, 2/1) that has an outer tag (customer VLAN tag) equal to 10 is mapped to SPB service 1.

- SPB service 2 (I-SID=501, BVLAN=4002) is assigned to SAPs 1/12:0 and 2/1:all. All tagged traffic (except for VLAN 10 tagged traffic, which is mapped to service 1) and untagged traffic received on the SAP access ports (1/12, 2/1) is mapped to SPB service 2.

Sample Command Configuration

This section provides the sequence of commands used on each switch to configure the "Sample SPBM Network Topology" on page 3-12. Note that the SPBM backbone is configured on every switch first, then the SPBM service architecture is configured second. Following this order of configuration is highly recommended to ensure proper switch participation in ISIS-SPB adjacencies and shortest path tree calculations.

SPBM Backbone Commands

The **system name** and Shortest Path Bridging (**spb**) commands are used to configure the SPBM backbone infrastructure for the sample topology, as shown:

BEB-1	BEB-2	BCB-1
-> system name BEB-1	-> system name BEB-2	-> system name BCB-1
-> spb bvlan 4001	-> spb bvlan 4001	-> spb bvlan 4001
-> spb bvlan 4002	-> spb bvlan 4002	-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1	-> spb isis bvlan 4001 ect-id 1	-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2	-> spb isis bvlan 4002 ect-id 2	-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001	-> spb isis control-bvlan 4001	-> spb isis control-bvlan 4001
-> spb interface port 1/1-3	-> spb interface port 1/1-3	-> spb interface port 1/1-4
-> spb isis admin-state enable	-> spb isis admin-state enable	-> spb isis admin-state enable

BCB-2	**BCB-3**	**BCB-4**
-> system name BCB-2	-> system name BCB-3	-> system name BCB-4
-> spb bvlan 4001	-> spb bvlan 4001	-> spb bvlan 4001
-> spb bvlan 4002	-> spb bvlan 4002	-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1	-> spb isis bvlan 4001 ect-id 1	-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2	-> spb isis bvlan 4002 ect-id 2	-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001	-> spb isis control-bvlan 4001	-> spb isis control-bvlan 4001
-> spb interface port 1/1-4	-> spb interface port 1/1-3	-> spb interface port 1/1-3
-> spb isis admin-state enable	-> spb isis admin-state enable	-> spb isis admin-state enable

BCB-5	**BCB-6**
-> system name BCB-5	-> system name BCB-6
-> spb bvlan 4001	-> spb bvlan 4001
-> spb bvlan 4002	-> spb bvlan 4002
-> spb isis bvlan 4001 ect-id 1	-> spb isis bvlan 4001 ect-id 1
-> spb isis bvlan 4002 ect-id 2	-> spb isis bvlan 4002 ect-id 2
-> spb isis control-bvlan 4001	-> spb isis control-bvlan 4001
-> spb interface port 1/1-3	-> spb interface port 1/1-3
-> spb isis admin-state enable	-> spb isis admin-state enable

SPBM Service Commands

The Service Manager (**service**) commands are used to build the SPBM services architecture for the sample topology, as shown. Note that services are only configured on designated BEB switches.

BEB-1	**BEB-2**
-> service access port 1/12	-> service access port 2/1
-> service 1 spb isid 500 bvlan 4001	-> service 1 spb isid 500 bvlan 4001
-> service 2 spb isid 501 bvlan 4002	-> service 2 spb isid 501 bvlan 4002
-> service 1 sap port 1/12:10 admin-state enable	-> service 1 sap 2/1:10 admin-state enable
-> service 2 sap port 1/12:0 admin-state enable	-> service 2 sap 2/1:0 admin-state enable
-> service 2 sap port 1/12:all admin-state enable	-> service 2 sap 2/1:all admin-state enable

Configuring SPBM

Configuring the SPBM backbone and service layers requires several steps. These steps are outlined here and further described throughout this section. For a brief tutorial on configuring SPBM, see "Quick Steps for Configuring SPBM" on page 3-25.

Configure the SPBM Backbone (ISIS-SPB)

Only switches that are SPB capable can participate in the SPBM network topology. The following configuration steps are required to make an OmniSwitch an SPB-capable node:

1 Create a BVLAN. The BVLAN provides the foundation of the SPBM infrastructure. A BVLAN is associated with an equal cost tree (ECT) algorithm ID and an SPB service instance ID that is used to carry customer traffic through the backbone network. See "Backbone VLANs" on page 3-29.

2 Configure SPB interfaces. An SPB interface is associated with each BVLAN that is configured on the switch. At the ISIS-SPB level, this type of interface sends and receives ISIS Hello packets and link state PDU (LSP) to discover adjacent SPB switches and calculate the shortest path trees through the SPBM network topology. At the services level, the SPB interfaces serve as network ports that are used to carry encapsulated customer traffic through the network. See "Configuring SPB Interfaces" on page 3-32.

3 Configure global ISIS-SPB parameters. In addition to enabling/disabling the ISIS-SPB instance for the switch, global configuration includes settings such as a system name for the switch, global bridge parameters, and various wait time intervals. When ISIS-SPB is enabled for the switch, default settings for these global bridge parameters and wait time intervals are active. It is only necessary to change these values if the default settings are not sufficient. See "Configuring Global ISIS-SPB Parameters" on page 3-34.

For more information about SPBM commands, see Chapter 9, "Shortest Path Bridging Commands," in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configure SPBM Services

The OmniSwitch Service Manager application is used to configure the services layer of the SPBM network topology. A service is defined by a specific set of logical entities that are configured only on the backbone edge bridges (BEBs) of the network. The following configuration steps are required to define a service-based architecture for an SPBM network:

1 Create an SPBM service. A Service Manager service ID is associated with a BVLAN, a backbone service instance identifier (I-SID), and a service access point (SAP) to identify the customer traffic that the service will tunnel through the provider network. See "Creating an SPB Service" on page 3-39.

2 Configure access (customer-facing) ports. One or more access ports are associated with a service access point (SAP) to identify to the service which ports will receive customer traffic that the service will process for tunneling through the provider network. When an access port is associated with a SAP, the SAP parameter attributes are applied to traffic received on the access port. See "Configuring Service Access Ports" on page 3-43.

3 Define access port profile attributes. A default Layer 2 profile is automatically assigned to an access port at the time the port is configured as an access port. This profile determines how control frames received on the port are processed. It is only necessary to configure a Layer 2 profile if the default attribute values are not sufficient. See "Configuring Layer 2 Profiles for Access Ports" on page 3-44.

4 Configure an SPB service access point (SAP). A SAP binds an SPB service to an access (customer-facing) port and defines which customer traffic to tunnel through the service. Each SAP is associated to one service name, but a single service can have multiple SAPs to which it is associated. See "Configuring Service Access Points (SAPs)" on page 3-42.

To define a Remote Fault Propagation (RFP) domain to monitor SPB services, see "Configuring Remote Fault Propagation for SPBM" on page 3-49.

For more information about Service Manager commands, see Chapter 10, "Service Manager Commands," in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

SPB Configuration Guidelines

Configuring an SPBM network topology involves setting up two layers of functionality: the ISIS-SPB backbone infrastructure and the Provider Backbone Bridge (802.1ah) services layer for MAC-in-MAC encapsulation. Review the guidelines in this section before attempting to configure the various components of the SPBM infrastructure and services.

ISIS-SPB

This implementation of the ISIS-SPB protocol supports only a single topology with a multi-topology identifier (MT-ID) of zero.

- The ISIS-SPB protocol instance is independent of IP IS-IS, or other network layer protocol identifiers (NLPIDs) riding in the same IS-IS implementation. However, ISIS-SPB and IP IS-IS can coexist on the same switch.

- ISIS-SPB interfaces, link state packet databases (LSPDB), and forwarding information are all created and maintained within the single ISIS-SPB instance.

- IS-IS Level 1 point-to-point adjacencies are supported; Level 2 is not supported at this time.

- SPB interfaces are associated with a link metric cost that is configurable, thus providing the ability to change the logical topology created by the ISIS-SPB instance. However, if different metric values are configured on each side of a link, ISIS-SPB will choose the higher-valued one as the metric to use for both sides. This is necessary to enforce the symmetry of SPT calculations in both directions across the link.

- Enabling SPB for the switch automatically triggers the transmission of Hello packets from the SPB interfaces, thus starting the process of discovery and forming adjacencies to build shortest path trees.

Backbone VLANs

- The BVLAN configuration must be the same on each SPB switch within the PBB network.For example, if BVLAN 10 with an ECT ID of 1 is configured on one switch, then BVLAN 10 with an ECT ID of 1 must exist on all other SPB bridges in the network to ensure proper calculation of the ISIS-SPB shortest path trees through the backbone.

- If more than one BVLAN is needed, configure each BVLAN with a different ECT algorithm ID. For example, if two BVLANs (BVLAN 4001 and BVLAN 4002) are needed for a specific SPBM topology, then create BVLAN 4001 with ECT ID 1 and BVLAN 4002 with ECT ID 2 on each switch that is going to participate in the topology.

> **Note.** When adding another BVLAN to an existing SPBM topology instance, create the new BVLAN and its associated ECT ID on every switch first, then configure the SPB service association for the BVLAN. Creating SPB services before the BVLAN configuration is complete on all switches can cause problems with forming adjacencies or may even cause an SPB switch to drop existing adjacencies.

- In most cases one BVLAN is sufficient for virtualizing traffic through the network backbone. However, configuring more than one BVLAN provides alternate routes for tunneling customer traffic. This can also provide a form of load balancing by distributing traffic over different BVLAN segments.

- All encapsulated traffic within the BVLAN domain is unicast with a resolved source and destination BMAC addresses. Frames received on BVLAN ports that do not have an incoming source MAC address pre-programmed by ISIS-SPB are discarded.

Configuring BVLANs

The SPBM backbone VLAN (BVLAN) provides the foundation on which ISIS-SPB shortest path trees are built and SPBM services tunnel encapsulated customer data through the Provider Backbone Bridge network (PBBN). Configuring a BVLAN on a switch is also the first step in setting up the ISIS-SPB infrastructure and in making an OmniSwitch an SPB-capable node.

> **Note.** The BVLAN configuration must be the same on each OmniSwitch that is going to participate in the SPBM network topology. So if BVLAN 4001 is created on one switch, then BVLAN 4001 must be created on all other switches in the SPBM network.

To create a BVLAN, use the **spb bvlan** command with the optional **name** parameter. For example:

```
-> spb bvlan 4001 name spb-4001
```

If the **name** parameter is not specified with this command, the VLAN ID is used for the name by default. For example, the following command creates BVLAN 4001 with "VLAN 4001" as the name:

```
-> spb bvlan 4001
```

To remove a BVLAN, use the **no** form of the **spb bvlan** command. For example:

```
-> no spb bvlan 4001
```

Assigning the Equal Cost Tree ID

ISIS-SPB calculations may result in multiple paths of equal costs. The Equal Cost Tree (ECT) ID specifies a tie-breaking algorithm that is used when ISIS-SPB is calculating a set of shortest path trees from one switch to all other switches in the SPB domain. When a BVLAN is created, an ECT ID is automatically assigned to the BVLAN. If it is the first BVLAN created on the switch, ECT ID 1 is assigned, otherwise the next available ID number is used.

Each BVLAN created must be duplicated on all other participating switches in the SPBM network and must use the same ECT ID number for that BVLAN on each switch. A BVLAN created on one switch may not be automatically assigned the same ECT ID on another switch. As a result, it may be necessary to modify the ECT ID number using the **spb isis bvlan ect-id** command. For example:

```
-> spb isis bvlan 4002 ect-id 2
```

Note. When adding another BVLAN to an existing SPBM topology instance, create the new BVLAN and its associated ECT ID on every switch first, then configure the SPB service association for the BVLAN. Creating SPB services before the BVLAN configuration is complete on all switches can cause problems with forming adjacencies or may even cause an SPB switch to drop existing adjacencies.

Configuring the Control BVLAN

One of the BVLANs configured on each switch serves as the control BVLAN for the ISIS-SPB instance. The control BVLAN exchanges ISIS-SPB control packets with neighboring SPB switches on behalf of all BVLANs configured on the local switch. The control packets are tagged with the control BVLAN ID.

To designate a BVLAN as the control BVLAN, use the **spb isis control-bvlan** command. For example:

```
-> spb isis control-bvlan 4002
```

A control BVLAN also carries regular encapsulated SPB domain traffic in addition to ISIS-SPB control packets. In other words, a VLAN can serve as both a regular BVLAN and a control BVLAN at the same time.

Configuring the Tandem Multicast Mode

The tandem multicast mode (*,G) or (S,G) of a BVLAN is applied only to SPB services associated with the BVLAN that are using tandem replication for multicast traffic. When a BVLAN is created, the (S,G) tandem multicast mode is applied by default.

To change the tandem multicast mode for a BVLAN, use the **spb isis bvlan tandem-multicast-mode** command and specify either **gmode** (*,G) or **sgmode** (S,G). For example:

```
-> spb isis bvlan 4001 tandem-multicast-mode sgmode
-> spb isis bvlan 4002 tandem-multicast-mode gmode
```

Verifying the BVLAN Configuration

To view the BVLAN configuration for the switch, use the **show spb isis bvlans** command. For example:

```
-> show spb isis bvlans
SPB ISIS BVLANS:

                                 Services  Num     Tandem      Root Bridge
 BVLAN   ECT-algorithm   In Use  mapped    ISIDS   Multicast  (Name : MAC Address)
-------+---------------+------+---------+------+----------+--------------------
  4001  00-80-c2-01      YES     YES        52   SGMODE
  4002  00-80-c2-02      YES     YES        51   SGMODE
  4003  00-80-c2-03      YES     YES        52   SGMODE
  4004  00-80-c2-04      YES     YES        51   SGMODE

BVLANs:     4
```

The BVLAN is a special type of VLAN that is created and maintained by VLAN Manager. As a result, it also appears in the VLAN Manager **show** command displays. For example, in the following **show vlan** output display, VLANs 4001 through 4004 are included and "spb" appears in the "type" column:

```
-> show vlan
 vlan  type  admin  oper  ip   mtu   name
-----+------+-----+-----+-----+-----+------------
    1  std   Dis   Dis   Dis  1500  VLAN 1
 1000  std   Ena   Ena   Ena  1500  VLAN 1000
 4001  spb   Ena   Ena   Dis  1524  VLAN 4001
 4002  spb   Ena   Ena   Dis  1524  VLAN 4002
 4003  spb   Ena   Ena   Dis  1524  VLAN 4003
 4004  spb   Ena   Ena   Dis  1524  VLAN 4004
 4094  mcm   Ena   Dis   Dis  9198  MCM IPC
```

To view configuration information for an individual BVLAN, use the **show vlan** command and specify the BVLAN ID. For example:

```
-> show vlan 4001
Name                     : VLAN 4001,
Type                     : Backbone vlan,
Administrative State     : enabled,
Operational State        : disabled,
IP Router Port           : disabled,
IP MTU                   : 1524
```

Configuring SPB Interfaces

A port or link aggregate is configurable as an SPB interface. Each switch in the SPBM topology should have at least one SPB interface configured. The SPB interface serves more than one purpose:

- Advertises IS-IS Hello packets to discover SPB neighbors and establish adjacencies.

- After adjacencies are established, exchanges link state packets (LSPs) with SPB neighbors to build a local LSP database (LSPDB). A switch's adjacencies are reflected in the contents of its link state packets. This relationship between adjacencies and link state allows the protocol to detect downed routers in a timely fashion.

- Serves as a network port by forwarding encapsulated SPB service traffic on backbone VLANs (BVLANs) through the SPBM Provider Backbone Bridge (PBB) network.

To configure a port or link aggregate as SPB interface, use the **spb isis interface** command. For example:

```
-> spb isis interface port 1/10
-> spb isis interface linkagg 5
```

When a port is converted to an SPB interface, the interface is automatically assigned to all existing BVLANs. There is one ISIS-SPB instance per switch, and each BVLAN and SPB interface are associated with that instance. However, it is also possible to tag SPB interfaces to carry traffic for standard VLANs.

The **spb isis interface** command is also used to optionally configure the following parameter values:

- **admin-state**—Administratively enables or disables the SPB interface. By default, the interface is enabled when the SPB interface is created.

- **hello-interval**—Specifies the amount of time, in seconds, to wait between each transmission of a hello packet from this interface. By default, the hello time interval is set to nine seconds.

- **hello-multiplier**—Specifies an integer value that is multiplied by the hello interval time to determine the amount of time, in seconds, a receiving bridge holds onto the hello packets transmitted from this interface. By default, the hello multiplier is set to three.

- **metric**—An integer value that specifies the link cost to reach the destination backbone MAC (BMAC), By default, the link cost is set to ten. Changing the link metric value provides a method for changing the logical topology as calculated by ISIS-SPB.

The following command examples change the default hello and metric values for the SPB interface:

```
-> spb isis interface port 4/7 hello-interval 60
-> spb isis interface linkagg 3 hello-multiplier 10
-> spb isis interface port 2/1 metric 100
-> spb isis interface linkagg 5 hello-interval 20 hello-multiplier 5 metric 200
```

Verifying the SPB Interface Configuration

To view the SPB interface configuration for the switch, use the **show spb isis interface** command. For example:

```
-> show spb isis interface
SPB ISIS Interfaces:
                                 Oper   Admin   Link     Hello   Hello
  Interface      Level   CircID  state  state   Metric   Intvl   Mult
---------------+-------+---------+------+-------+---------+-------+-------
  1/1            L1      1        UP     UP      10       9       3
  1/2            L1      2        UP     UP      10       9       3
  1/3            L1      3        UP     UP      10       9       3
  1/4            L1      4        DOWN   UP      10       9       3
  1/5            L1      5        DOWN   UP      10       9       3
  1/6            L1      6        DOWN   UP      22       9       3
  1/7            L1      7        DOWN   UP      10       9       3

Interfaces : 7
```

Configuring Global ISIS-SPB Parameters

This section describes the global configuration for the ISIS-SPB instance, which includes the following:

- "Configuring the System Name" on page 3-35.

- "Configuring the SPB Bridge Priority" on page 3-35.

- "Configuring the ISIS-SPB Area Address" on page 3-35.

- "Configuring the Shortest Path Source ID" on page 3-35.

- "Configuring the Control MAC Address" on page 3-35.

- "Configuring the Shortest Path First Wait Time" on page 3-36.

- "Configuring the Link State Packet Wait Time" on page 3-37.

- "Configuring the Overload State" on page 3-37.

- "Configuring Redundant Switches for Graceful Restart" on page 3-38.

- "Enabling/Disabling ISIS-SPB" on page 3-39.

To verify the global configuration parameter values for the switch, use the **show spb isis info** command. For example:

```
-> show spb isis info
SPB ISIS Bridge Info:
  System Id            = e8e7.3233.1831,
  System Hostname      = BEB-1,
  SPSourceID           = 03-18-31,
  SPBM System Mode     = auto,
  BridgePriority       = 32768 (0x8000),
  MT ID                = 0,
  Control BVLAN        = 4001,
  Area Address         = 0.0.0,
  Level Capability     = L1,
  Admin State          = UP,
  LSDB Overload        = Disabled,
  Last Enabled         = Thu Aug  2 22:43:19 2012,
  Last SPF             = Fri Aug  3 18:15:51 2012,
  SPF Wait             = Max: 1000 ms, Initial: 100 ms, Second: 300 ms,
  LSP Lifetime         = 1200,
  LSP Wait             = Max: 1000 ms, Initial: 0 ms, Second: 300 ms,
  Graceful Restart     = Disabled,
  GR helper-mode       = Disabled,
  # of L1 LSPs         = 8
  Control Address      = 01:80:C2:00:00:14 (AllL1)
```

Configuring the System Name

Configuring a system name is required on each switch that is going to participate in the SPBM topology. To configure a system name for the switch, use the **system name** command. For example:

```
-> system name BEB-1
```

ISIS-SPB advertises the system name to identify the switch to other SPB peer switches.

Configuring the SPB Bridge Priority

A bridge is ranked within the SPB topology by its bridge ID (an eight byte hex number). The bridge priority value makes up the upper two bytes of the eight-byte SPB bridge ID. The lower six bytes of the Bridge ID contain the system ID, which is the dedicated bridge MAC address of the SPB bridge.

The bridge priority is used in shortest path tree calculations. The lower the priority value, the higher the priority. Setting a different bridge priority value on different SPB bridges will override the system ID significance during the shortest path tree (SPT) calculation.

By default, all SPB switches are assigned a priority value of 32768. To change the bridge priority value for a switch, use the **spb isis bridge-priority** command. For example:

```
-> spb isis bridge-priority 25590
```

Configuring the ISIS-SPB Area Address

By default, the IS-IS area address for the ISIS-SPB instance is set to 0.0.0, which is typically sufficient for this implementation of SPBM. Both ISIS-SPB and ISIS-IP instances may coexist on the same switch as long as they don't use the same area address.

If changing the area address is necessary, use the **spb isis area-address** command. For example:

```
-> spb isis area-address 1.1.1
```

Note. Each switch that is going to participate in the SPB topology must use the same area address and must use an address that is different from the ISIS-IP area address.

Configuring the Shortest Path Source ID

The shortest path (SP) source ID, identifies the source of multicast frames and is relevant only in multicast tandem replication mode. By default, the last three least significant bytes of the system ID (local bridge MAC address) is used for the source ID value.

To change the source ID value, use the **spb isis source-id** command. For example:

```
-> spb isis source-id 07-0b-d3
```

To set the source ID back to the default value, use the **spb isis source-id** command with the **auto** parameter. For example:

```
-> spb isis source-id auto
```

Configuring the Control MAC Address

The control MAC address is the destination MAC address used for ISIS-SPB control packets. Changing this address can enhance interoperability between an SPB-capable OmniSwitch and other third-party SPB-capable devices.

By default, the control MAC address is set to 01:80:C2:00:00:14 (all Level 1 Intermediate Systems). The following parameters are used with the **spb isis control-address** command to change the control MAC address:

- **alll1**—All Level 1 Intermediate Systems (01:80:C2:00:00:14).

- **alll2**—All Level 2 Intermediate Systems (01:80:C2:00:00:15).

- **allis**—All Intermediate Systems (09:00:2B:00:00:05).

For example, the following command changes the default control MAC address from AllL1 to AllL2:

```
-> spb isis control-address alll2
```

Configuring the Shortest Path First Wait Time

The **spb isis spf-wait** command is used to configure the time intervals between the first, second, and subsequent ISIS-SPB shortest path first (SPF) calculations.

Subsequent SPF calculations, if required, are generated at exponentially increasing intervals of the SPF second wait time interval until the maximum wait time interval value is reached. For example, if the second-wait interval value is set to 1000 milliseconds, then the next SPF calculation is triggered after 2000 milliseconds and the next SPF calculation after that is triggered at 4000 milliseconds, and so on, until the maximum-wait interval value is reached.

When the maximum interval value is reached, the SPF wait interval will stay at the maximum value until there are no more SPF calculations scheduled during that interval. After a full interval without any SPF calculations, the SPF wait interval will reset back to the initial wait time interval value.

The following **spb isis spf-wait** command parameters are used to configure the SPF timers:

- **max-wait**—The maximum number of milliseconds to wait between two consecutive SPF calculations. The default maximum wait time value is set to 1000 milliseconds. Specify a maximum value that is the same or greater than the second wait time value.

- **initial-wait**—The number of milliseconds to wait before triggering an initial SPF calculation after a topology change. The default initial wait time value is set to 100 milliseconds. Specify a value that is the same or less than the maximum wait time value.

- **second-wait**—The number of milliseconds to wait between the first and second SPF calculation. The default second wait time value is set to 300 milliseconds. Specify a value that is the same or less than the maximum wait time value.

For example, the following command changes the SPF wait time values for the local SPB instance:

```
-> spb isis spf-wait max-wait 2500 initial-wait 1000 second-wait 1500
```

To change one or more of the wait time values, it is only necessary to specify the parameter for the desired change. For example:

```
-> spb isis spf-wait max-wait 5000
-> spb isis spf-wait initial-wait 1000
-> spb isis spf-wait second-wait 2000
```

To set the wait time values back to the default settings. use the **spb isis spf-wait** command without specifying any of the parameters. For example:

```
-> spb isis spf-wait
```

Configuring the Link State Packet Wait Time

The **spb isis lsp-wait** command is used to configure the time intervals between the first, second, and subsequently generated link state packets (LSPs).

Subsequent LSP, if required, are generated at exponentially increasing intervals of the LSP second wait time interval until the maximum value is reached. For example, if the second-wait interval value is set to 10 seconds, then the next LSP is generation is triggered after 20 seconds and the next LSP generated after that is triggered at 40 seconds, and so on, until the maximum wait time interval value is reached.

When the maximum interval value is reached, the LSP wait interval will stay at the maximum value until there are no more LSP generations during that interval. After a full interval without any LSP generations, the LSP wait interval will reset back to the initial wait time interval value.

The following **spb isis lsp-wait** command parameters are used to configure the SPF timers:

- **max-wait**—The maximum number of seconds to wait between two consecutively generated LSPs. The default maximum wait time value is set to 1000 milliseconds. Specify a maximum value that is the same or greater than the second wait time value.

- **initial-wait**—The number of seconds to wait before triggering an initial LSP generation after a topology change. The default initial wait time value is set to 0 milliseconds. Specify a value that is the same or less than the maximum wait time value.

- **second-wait**—The minimum number of seconds to wait between the first and second generated LSPs. The default second wait time value is set to 300 milliseconds. Specify a value that is the same or less than the maximum wait time value.

For example, the following command changes the LSP wait time values for the local SPB instance:

```
-> spb isis lsp-wait max-wait 2000 initial-wait 1000 second-wait 1500
```

To change one or more of the wait time values, it is only necessary to specify the parameter for the desired change. For example:

```
-> spb isis lsp-wait max-wait 5000
-> spb isis lsp-wait initial-wait 2500
-> spb isis lsp-wait second-wait 3000
```

To set the wait time values back to the default settings. use the **spb isis lsp-wait** command without specifying any of the parameters. For example:

```
-> spb isis lsp-wait
```

Configuring the Overload State

This implementation of ISIS-SPB supports the overload state mechanism, which allows an instance of ISIS-SPB to inform its neighbors that the instance is nearing or exceeding its capabilities. When peers see that a switch is advertising in this state, they will select an alternate path around the overloaded switch.

The ISIS-SPB instance for a switch may dynamically trigger the overload state condition when the instance detects that it is nearing or has reached resource limits. However, it is also possible to manually trigger the overload state condition using the **spb isis overload** command. For example:

```
-> spb isis overload
```

Some advantages of manually triggering the overload state condition, even if the instance is no where near its resource limits, are as follows:

- The switch is designated as "leaf node" that should never carry transit traffic. Configuring the link metric value for the SPB interfaces on the switch and attached peers is another method for preventing the switch from receiving transit traffic, but enabling the overload state is a much quicker way to achieve the same results and requires less configuration.

- When there is a need to remove the switch from service (temporarily or permanently). In this scenario, network availability is increased because peer switches will detect the overload state of the switch and gracefully transition to alternate paths, while the "manually overloaded" switch continues to forward packets. Just simply shutting the switch down would cause more disruption to network traffic.

When the overload state is either dynamically or manually enabled for the switch, the overload bit is set in LSP 0 to indicate that this ISIS-SPB instance is not available to accept transit traffic. However, an ISIS-SPB switch operating in the overload state is still used only if there is no alternate path to reach the intended destination.

When the overload state is enabled, the switch will operate in this state for an infinite amount of time. To configure the switch to remain in the overload state for only a specific amount of time (in seconds), use the **spb isis overload** command with the optional **timeout** parameter. For example:

```
-> spb isis overload timeout 500
```

To disable the overload state, use the **no** form of the **spb isis overload** command. For example:

```
-> no spb isis overload
```

It is also possible to specify that the overload state is enabled for the switch after every system bootup. This is done using the **spb isis overload-on-boot** command, which also has an optional **timeout** parameter. For example:

```
-> spb isis overload-on-boot timeout 500
```

To disable the overload-on-boot option, use the **no** form of the **spb isis overload-on-boot** command. For example:

```
-> no spb isis overload-on-boot timeout 500
```

Note that the **no spb isis overload** command does not disable the overload-on-bootup option.

Configuring Redundant Switches for Graceful Restart

By default, ISIS-SPB graceful restart is enabled. When graceful restart is enabled, the switch can either be a helper (which helps a neighbor router to restart) or a restarting router, or both. When graceful restart is enabled on the switch, the helper mode is automatically enabled by default.

To configure ISIS-SPB graceful restart support on an OmniSwitch, use the **spb isis graceful-restart** command. For example, to configure graceful restart on the router, enter:

```
-> spb isis graceful-restart
```

The helper mode can be disabled on the switch with the **spb isis graceful-restart helper** command. For example, to disable the helper support for neighboring switches, enter the following:

```
-> ip isis graceful-restart helper disable
```

To disable support for graceful restart, use the **no** form of the **spb isis graceful-restart** command. For example:

```
-> no spb isis graceful-restart
```

Enabling/Disabling ISIS-SPB

By default ISIS-SPB is disabled on the switch. To enable ISIS-SPB, use the **spb isis admin-state** command with the **enable** option. For example:

```
-> spb isis admin-state enable
```

To disable the ISIS-SPB instance on the switch, enter the **spb isis admin-state** command with the **disable** option. When the ISIS-SPB status s disabled for the switch, the related configuration settings and statistics are retained.

```
-> spb isis admin-state disable
```

Note. Enabling ISIS-SPB on a switch starts the process of ISIS-SPB discovery, adjacency building, and shortest path tree calculations. Make sure that the SPBM configuration is set up first, then enable ISIS-SPB on each switch that will participate in the SPBM network.

Creating an SPB Service

An SPB service is identified by a service ID number, which represents an association between a backbone service instance identifier (I-SID) and an existing BVLAN. Basically, creating an SPB service binds the backbone I-SID to a BVLAN ID. All traffic mapped to the specific I-SID is then encapsulated and forwarded on the associated BVLAN to the intended destination.

The **service spb** command is used to create an SPB service. For example, the following command creates SPB service 1 and binds I-SID 100 to BVLAN 4001:

```
-> service 1 spb isid 500 bvlan 4001 admin-state enable
```

The BVLAN ID specified with the **service spb** command must already exist in the switch configuration. However, the I-SID number specified creates a new I-SID that is bound to the BVLAN for this service.

Note. When adding another BVLAN to an existing SPBM topology instance, create the new BVLAN and its associated ECT ID on every switch first, then configure the SPB service association for the BVLAN. Creating SPB services before the BVLAN configuration is complete on all switches can cause problems with forming adjacencies or may even cause an SPB switch to drop existing adjacencies.

Modifying Default SPB Service Parameters

The following SPB service parameter values are set by default at the time the service is created. If necessary, use the specified commands to change the default values.

Parameter Description	Command	Default
Service description.	**service description**	None
Administrative status for statistics collection.	**service stats**	Disabled
Multicast replication mode	**service multicast-mode**	head-end
VLAN translation	**service vlan-xlation**	Disabled
Administrative status of the service	**service admin-state**	Disabled

Refer to the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the above parameters and related commands.

Using VLAN Translation

VLAN translation refers to the egress translation of VLAN tags on service access points (SAPs). When enabled for a service, the VLAN tags for outgoing customer frames on SAPs associated with that service are processed according to the local SAP configuration (the SAP on which the frames will egress) and not according to the configuration of the SAP on which the frames were received.

- If the local SAP is configured for untagged traffic (*slot/port*:0), the egress traffic is always sent out as untagged.

- If the local SAP is configured for 802.1q-tagged traffic (*slot/port:ctag*), the egress traffic is single-tagged with the tag value specified by the *ctag* (customer VLAN tag) value.

- If the local SAP is configured for double-tagged traffic (*slot/port:outer_tag.: inner_tag*), the egress traffic is double-tagged with the tag values specified by the *outer_tag* and *inner_tag* values.

When VLAN translation is disabled, frames simply egress without any modification of the VLAN tags. In other words, the frames are transparently bridged without tag modification.

The following table shows the required translation (tag is added or replaced) that takes place when the egress SAP configuration is applied to the possible frame types (untagged, tagged, double-tagged). Note that in this table the terms "ITAG" and "OTAG" refer to inner tag and outer tag, respectively.

	Egress SAP (action required based on SAP type)		
	Untagged SAP	**Single Tagged SAP**	**Double-Tagged SAP**
Incoming Frame	Remove OTAG Remove ITAG	Replace OTAG **Note**: Replace = implicit add Remove ITAG	Replace OTAG **Note**: Replace = implicit add Add/Replace ITAG
Untagged	No tags, no action taken	Add the SAP OTAG	Add the SAP OTAG Add the SAP ITAG.
Single-tagged	Remove the OTAG	Replace the OTAG	Add ITAG Replace OTAG
Double-tagged	Remove the ITAG Remove the ITAG	Remove the ITAG Replace the OTAG	Replace ITAG Replace OTAG

Enabling VLAN translation is required at two different levels: first at the access port level and then at the service level. This activates VXLAN translation for all of the SAPs on an access port that belong to the same service.

To enable translation at the service level, use the **service vlan-xlation** command. For example:

```
-> service 1 vlan-xlation enable
```

To enable VLAN translation for all services, use the **all** parameter with the same command. For example:

```
-> service all vlan-xlation enable
```

To disable VLAN translation, use the **service vlan-xlation** command with the disable parameter. For example:

```
-> service 1 vlan-xlation disable
-> service all vlan-xlation disable
```

To enable VLAN translation at the access port level, use the **service access vlan-xlation** command. For example:

```
-> service access port 1/11 vlan-xlation enable
```

See "Configuring Service Access Ports" on page 3-43 for more information.

Enable the Service

By default, the SPB service is disabled when the service is created. Once the service is created and any optional service parameters are configured, use the **service admin-state** command with the **enable** option to enable the service. For example:

```
-> service 1 admin-state enable
```

To disable the service, enter the following command:

```
-> service 1 admin-state disable
```

Deleting an SPB Service

Before deleting a service from the switch configuration, disable the administrative status of the service. Once this is done, use the **no** form of the **service spb** command to delete the service. For example:

```
-> no service 1 spb
```

Verifying the SPB Service Configuration

To view the SPB service configuration for the switch, use the **show service** command with the **spb** parameter option. For example:

```
-> show service spb
Legend: * denotes a dynamic object
SPB Service Info
  SystemId : 00e0.b1e7.0188,   SrcId : 0x70188,    SystemName : BEB-1


                          SAP     Bind                MCast
ServiceId  Adm  Oper Stats Count   Count   Isid      BVlan Mode      (T/R)
-----------+----+----+-----+-------+-------+---------+-----+--------------
1          Up   Up   N    4       1       1000      4001  Headend   (0/0)
2          Up   Up   N    4       1       1001      4001  Headend   (0/0)
3          Up   Up   N    4       1       1002      4001  Headend   (0/0)
4          Up   Up   N    4       1       1003      4001  Headend   (0/0)
5          Up   Up   N    4       1       1004      4001  Headend   (0/0)
6          Up   Up   N    4       1       1005      4001  Headend   (0/0)
7          Up   Up   N    4       1       1006      4001  Headend   (0/0)
8          Up   Up   N    4       1       1007      4001  Headend   (0/0)
9          Up   Up   N    4       1       1008      4001  Headend   (0/0)
10         Up   Up   N    4       1       1009      4001  Headend   (0/0)
```

To view the configuration for an individual service, use the **show service spb** command and specify the SPB service ID. For example:

```
-> show service spb 1
SPB Service Detailed Info
  Service Id       : 1,                    Description      : ,
  ISID             : 1000,                 BVlan            : 4001,
  Multicast-Mode   : Headend,              Tx/Rx Bits       : 0/0,
  Admin Status     : Up,                   Oper Status      : Up,
  Stats Status     : No,                   Vlan Translation : No,
  Service Type     : SPB,                  Allocation Type  : Static,
  MTU              : 9194,                 Def Mesh VC Id   : 1,
  SAP Count        : 4,                    SDP Bind Count   : 1,
  Ingress Pkts     : 0,                    Ingress Bytes    : 0,
  Egress Pkts      : 0,                    Egress Bytes     : 0,
  Mgmt Change      : 08/10/2012 13:14:43,  Status Change    : 08/10/2012 13:14:00
```

Configuring Service Access Points (SAPs)

A SAP identifies the location where customer traffic enters the Provider Backbone Bridge Network (PBBN) edge, the type of customer traffic to service, parameters to apply to the traffic, and the service that will process the traffic for tunneling through the provider network.

Configuring a SAP requires several steps. These steps are outlined here and further described throughout this section:

- Configure customer-facing ports or link aggregates as service access ports.

- Configure Layer 2 profiles to determine how control packets are processed on access ports.

- Create a SAP by associating a SAP ID with an SPB service ID. A SAP ID is comprised of an access port and an encapsulation value, which is used to identify the type of customer traffic (untagged, single-tagged, or double-tagged) to map to the associated service.

SAP Configuration Guidelines

Consider the following when configuring a SAP:

- A SAP is a unique local entity for any given device. The same SAP ID value can be used on other BEB switches.

- There are no SAPs configured by default; explicit configuration of a SAP is required.

- A SAP is administratively disabled at the time the SAP is created.

- When a SAP is deleted, all configuration parameters for the SAP are also deleted.

- A SAP is owned by and associated with the service that was specified at the time the SAP was created.

- Multiple SAPs with different service types, such as a Virtual eXtensible LAN (VXLAN) or an SPB service, are allowed on the same service access port. For example, the following **show service access** command output shows two SAPs for port 2/1/30: one SAP bound to a VXLAN service and the other SAP bound to an SPB service:

```
-> show service access port 2/1/30 sap
Legend: * denotes a dynamic object
                                                   Vlan
Identifier    Adm  Oper Stats T:P  ServiceId   Isid/Vnid Xlation Sap Description
------------+----+----+-----+----+----------+---------+-------+--------------
sap:2/1/30A:0 Down Down  N    Y:x  20          1500        N       -
sap:2/1/30A:5 Up   Down  N    Y:x  10          23000       N       -

Total SAPs: 2
```

- If a port is administratively shutdown, all SAPs on that port become operationally out of service.

- Both fixed ports and link aggregates are configurable as access ports. Only access ports are associated with SAPs.

- Bridging functionality is not supported on access ports or link aggregates.

- Configuring multiple SAPs on an access port that map different VLAN tags to the same service can cause a MAC move when the same customer MAC (CMAC) ingresses the access port with different VLAN tags. For example, a CMAC has two flows tagged with VLAN 10 and VLAN 20 ingressing access port 1/1 and both are mapped to service 100.

```
-> service 100 sap port 1/1:10
-> service 100 sap port 1/1:20
```

To avoid the MAC move in this scenario, use one of the following alternative SAP configurations.

Configure the SAPs with different services:

```
-> service 100 sap port 1/1:10
-> service 200 sap port 1/1:20
```

Configure a default SAP to classify both flows into the same service:

```
-> service 100 sap port 1/1:all
```

See for more information.

Configuring Service Access Ports

Each SAP is comprised of an access port or link aggregate and an encapsulation type value. Access ports are customer-facing ports that reside on a provider edge router. Traffic received on these ports is classified for one or more SAPs and forwarded onto the intended destination by the associated SPB service.

To configure a port or link aggregate as an access port, use the **service access** command. For example, the following command configures port 1/2 and link aggregate 5 as access ports:

```
-> service access port 1/2
-> service access linkagg 5
```

To revert an access port back to a regular switch port or link aggregate, use the no form of the service access command. For example:

```
-> no service access port 1/2
-> no service access linkagg 5
```

VLAN Translation on Access Ports

VLAN translation refers to the egress translation of VLAN tags on service access points (SAPs). For more information about how VLAN translation is applied, see "Using VLAN Translation" on page 3-40.

By default, VLAN translation is disabled on access ports. Enabling VLAN translation on an access port implicitly enables translation for all SAPs associated with that port. However, translation must also be enabled for the services associated with these SAPs. This ensures that all SAPs associated with a service will apply VLAN translation.

To enable VLAN translation on an access port, use the **service access vlan-xlation** command with the **enable** option. For example:

```
-> service access port 1/3 vlan-xlation enable
-> service access linkagg 10 vlan-xlation enable
```

To disable VLAN translation on an access port, use the **service access vlan-xlation** command with the **disable** option. For example:

```
-> service access port 1/3 vlan-xlation disable
-> service access linkagg 10 vlan-xlation disable
```

Configuring Layer 2 Profiles for Access Ports

A Layer 2 profile determines how control frames ingressing on an access port are processed. When a port is configured as an access port, a default Layer 2 profile (**def-access-profile**) is applied to the port with the following default values for processing control frames:

Protocol	Default
STP	tunnel
802.1x	drop
802.3ad	peer
802.1ab	drop
GVRP	tunnel
AMAP	discard
MVRP	tunnel

If the default profile values are not sufficient, use the **service l2profile** command with the **tunnel**, **drop**, and **peer** options to create a new profile. For example, the following command creates a profile named "DropL2":

```
-> service l2profile DropL2 stp drop gvrp drop 802.1ab drop
```

Consider the following when configuring Layer 2 profiles:

- When a profile is created, the new profile inherits the default profile settings for processing control frames. The default settings are applied with the new profile unless they are explicitly changed. For example, the profile "DropL2" was configured to discard STP, GVRP, and 802.1ab frames. No other protocol settings were changed, so the default settings still apply for the other protocols.

- Remove any profile associations with access ports before attempting to modify or delete the profile.

- Not all of the control protocols are currently supported with the **peer**, **tunnel**, and **drop** parameters. Use the following table to determine the parameter combinations that are supported:

Protocol	Reserved MAC	peer	drop	tunnel
STP	01-80-C2-00-00-00	no	yes	yes
802.1x	01-80-C2-00-00-03	no	yes	yes
802.1ab	01-80-C2-00-00-0E	yes	yes	yes
802.3ad	01-80-C2-00-00-02	yes	no	no
GVRP	01-80-C2-00-00-21	no	yes	yes
MVRP	01-80-C2-00-00-21	no	yes	yes
AMAP	00-20-DA-00-70-04	yes	yes	no

To delete a Layer 2 profile, use the **no** form of the **service l2profile** command. For example, the following command deletes the "DropL2" profile:

```
-> no service l2profile DropL2
```

Use the **show service l2profile** command to view a list of profiles that are already configured for the switch. This command also displays the attribute values for each profile.

Assigning Layer 2 Profiles to Access Ports

After a Layer 2 profile is created, it is then necessary to assign the profile to an access port or link aggregate. When this is done, the current profile associated with the port is replaced with the new profile.

The **service access l2profile** command is used to assign a new profile to an access port. For example, the following command assigns the "DropL2" profile to access port 1/4 and link aggregate 5:

```
-> service access port 1/4 l2profile DropL2
-> service access linkagg 5 l2profile DropL2
```

To change the profile associated with the access port back to the default profile (**def-access-profile**), use the **default** option with the **service access l2profile** command. For example:

```
-> service port 1/4 l2profile default
-> service access linkagg 5 l2profile default
```

Use the **show service access** command to display profile associations for access ports.

Verifying the Access Port Configuration

To view the access port configuration for the switch, use the **show service access** command. For example:

```
-> show service access
Port         Link  SAP     SAP     Vlan
Id           Status Type   Count   Xlation L2Profile
---------+------+-------+-------+-------+------------------
1/3         Up    Manual  100     N       def-access-profile
1/4         Down  Manual  100     N       def-access-profile
1/5         Down  Manual  100     N       def-access-profile
1/15        Down  Dynamic 0       Y       def-access-profile
1/16        Up    Dynamic 1       Y       def-access-profile
1/17        Down  Dynamic 0       Y       def-access-profile

Total Access Ports: 6
```

Creating the Service Access Point

Each SPB service is bound to at least one Service Access Point (SAP). A SAP identifies the point at which customer traffic enters the Provider Backbone Bridge Network (PBBN). Creating a SAP on an SPB switch designates that switch as a Backbone Edge Bridge (BEB) in the PBBN. An SPB switch that does not have a SAP but does have at least one BVLAN and an SPB interface is designated as Backbone Core Bridge (BCB) in the PBBN.

Once the SPB topology is determined and switches that will serve as BEBs are identified, a SAP is created on each BEB. A SAP is created by associating a SAP ID with an SPB service. A SAP ID is comprised of a customer-facing port (referred to as an access port) and an encapsulation value that is used to identify the type of customer traffic (untagged, single-tagged, or double-tagged) to map to the associated service.

The **service sap** command is used to configure a SAP. This command specifies the SPB service ID number and the SAP ID (slot/port:encapsulation). The following parameter values are used with this command to specify the encapsulation value:

SAP Encapsulation Value	Customer Traffic Serviced
0 (null)	All untagged packets; tagged packets are dropped.
all	All tagged and untagged packets not already classified into another SAP*
qtag	Only traffic 802.1q-tagged with the specified VLAN ID.
outer_*qtag.inner_qtag*	Only traffic double-tagged (QinQ) with the specified outer and inner VLAN IDs.

*Note that the **:all** (wildcard) parameter is also configurable as the inner tag value for double-tagged frames (for example, "10:all" specifies double-tagged packets with an outer tag equal to 10 and an inner tag with any value).

The following **service sap** command example creates a SAP that will direct customer traffic ingressing on access port 1/4 that is tagged with VLAN ID 50 to service 100:

```
-> service 100 sap 1/4:50 description "BEB1 to SPB100 CVLAN 50"
```

In the above example, the 1/4:50 designation is referred to as the SAP ID or the encapsulation ID. This means that if no other SAPs are configured for port 1/4, then any traffic ingressing on that port is dropped if the traffic is not tagged with VLAN 50.

It is possible to configure more than one SAP for the same access port, which provides a method for segregating incoming traffic into multiple services. For example, the following SAP configuration for port 2/3 sends incoming traffic to three different services based on the VLAN tags of the frames received:

```
-> service 2000 sap port 2/3:all
-> service 200 sap port 2/3:100
-> service 1000 sap port 2/3:100.200
```

In this example,

- Frames double-tagged with 100 (outer tag) and 200 (inner tag) are sent on service 1000.

- Frames single-tagged with VLAN 100 are sent on service 200.

- All other frames (those that are not single-tagged with 100 or double-tagged with 100 and 200) are sent on service 2000.

The following SAP ID classification precedence is applied when there are multiple SAPs for one access port:

1 Double-tagged (Outer VLAN + Inner VLAN) - Highest

2 Double-tagged (Outer VLAN + all)

3 Single-tagged (VLAN)

4 Single-tagged (wildcard)

5 Untagged - Lowest.

Modifying Default SAP Parameters

The following parameter values are set by default at the time the SAP is created. If necessary, use the specified commands to change the default values.

Parameter Description	Command	Default
SAP description.	**service sap description**	None
SAP trust mode	**service sap trusted**	Trusted
Administrative status for the SAP	**service sap admin-state**	Enabled
Administrative status for statistics collection.	**service sap stats**	Disabled

Refer to the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the command parameters.

Configuring the SAP Trust Mode

The **service sap trusted** command is used to configure the trust mode for a SAP. A trusted SAP can accept 802.1p values in incoming packets; an untrusted SAP will set any 802.1p values to zero in incoming packets, unless an 802.1p value is configured with this command.

Note that untagged Layer 2 control packets (for example, BPDU, GVRP, and AMAP) are always tunneled (if enabled) through the Provider Backbone Bridge (PBB) network with the default EXP bits set to 7, so that they can arrive at the destination bridge at the highest COS queue of 7. As a result, trusted and untrusted SAPs configured on the access ports will not affect the Layer 2 control packets ingressing on the access ports.

By default, a SAP is trusted with the priority set to best effort (zero). Use the **no** form of the **service spb sap trusted** command with the **priority** option to change the SAP mode to untrusted. For example:

```
-> service 100 sap 1/4:50 no trusted priority 7
```

When a SAP is trusted, the priority value contained in tagged customer packets is used; untagged packets are assigned the default priority value (zero). When a SAP is untrusted, the priority value configured for the SAP is assigned to both tagged and untagged customer packets.

Enabling/Disabling the SAP

By default, a SAP is disabled at the time the SAP is created. To enable the SAP administrative status, use the **service sap admin-state** command. For example:

```
-> service 100 sap port 1/4:50 admin-state enable
-> service 200 sap linkagg 5:all admin-state enable
```

To disable the SAP, enter the following command:

```
-> service 100 sap port 1/4:50 admin-state disable
-> service 200 sap linkagg 5:all admin-state disable
```

Deleting the SAP

When a SAP is administratively disabled, the SAP configuration is not removed from the switch. To delete a SAP from the switch configuration, use the **no** form of the **service sap** command. For example:

```
-> service 100 no sap port 1/4:50
-> service 200 no sap linkagg 5:all
```

Verifying the SAP Configuration

A SAP is a type of virtual port that is associated with an SPB service. To determine the SAP configuration for a specific service, use the **show service ports** command to view the virtual ports associated with a specific service. For example:

```
-> show service 1 ports
Legend: * denotes a dynamic object
SPB Service Info
  Admin : Up, Oper  : Up, Stats       : N, Mtu    : 9194,   VlanXlation : N,
  ISID  : 1000,      BVlan : 4001,   MCast-Mode : Headend,  Tx/Rx   : 0/0


                               Sap Trusted:Priority/       Sap Description /
Identifier       Adm  Oper Stats Sdp SystemId:BVlan   Intf    Sdp SystemName
----------------+----+----+-----+-------------------+--------+-------------------
sap:1/11:1000   Up   Up   N          Y:x            1/11       -
sap:1/12:1000   Up   Down N          Y:x            1/12       -
sap:1/13:1000   Up   Down N          Y:x            1/13       -
sap:1/14:1000   Up   Down N          Y:x            1/14       -
sdp:32776:1*    Up   Up   Y    e8e7.3233.1831:4001  1/1        BEB-1

Total Ports: 5
```

To then view configuration information for a specific SAP, use the **show service spb sap** command. For example:

```
-> show service 1 sap port 1/11:1000
SAP Detailed Info
  SAP Id        : 1/11:1000,       Description    : ,
  Admin Status  : Up,              Oper Status    : Up,
  Stats Status  : No,              Vlan Translation : No,
  Service Type  : SPB,             Allocation Type : Static,
  Trusted       : Yes,             Priority       : 0,
  Ingress Pkts  : 0,               Ingress Bytes  : 0,
  Egress Pkts   : 0,               Egress Bytes   : 0,
  Mgmt Change   : 08/07/2012 23:39:29,  Status Change  : 08/10/2012 15:13:08
```

Configuring Remote Fault Propagation for SPBM

Remote Fault Propagation (RFP) for SPBM monitors SPB access ports to detect link failures that cause interruptions to SPB services. The status of an access port and any associated I-SID is advertised within an RFP domain using Continuity Check Message (CCM) packets. When a CCM packet is received that indicates an access port for a specific I-SID is down, the corresponding access port associated with the same I-SID is automatically taken down.

Consider the following recommended guidelines when configuring RFP for SPBM:

- Configuring an RFP domain involves creating a local Maintenance End Point (MEP) on each switch that will participate in the RFP domain. The MEP is mapped to a reserved Ethernet OAM domain. This type of domain counts towards the maximum limit of Ethernet OAM domains allowed.

- The SPB control BVLAN serves as the primary VLAN for all RFP domains. CCM packets are sent across the SPBM network to all BEB devices on the control BVLAN. However, CCM packets are not encapsulated with SPB header information.

- Make sure to use the same CCM interval value for all local MEPs that participate in the same RFP domain. A mismatch will prevent reliable communication between MEPs.

- The SPB service associated with the I-SID that RFP will monitor should be configured on only two Backbone Edge Bridges (BEBs) in the network.

- The SPB service associated with an I-SID is mapped to only one SAP. For example, SPB service 10 bound to I-SID 1500 is mapped only to a SAP configured on port 1/12; service 10 is not mapped to any other SAP on the same switch.

- Configure only one SAP on a physical access port; configuring additional SAPs on the same port is not recommended.

- Configure a SAP associated with an RFP monitored port on only one physical port of the BEB.

Configuring an RFP Domain

A local Maintenance End Point (MEP) and a corresponding remote end point list are configured on each BEB that will participate in an RFP domain. The domain to which each end point is assigned is determined by the RFP domain ID associated with each end point. The domain ID is defined at the time the local MEP is created using the **service rfp local-endpoint** command. For example:

```
-> service rfp 1 local-endpoint 10 type spb
```

In this example, RFP domain 1 is created with a local MEP ID of 10. By default, the CCM interval is set to 1 second, the domain level is set to 7, and the administrative status is enabled for RFP domain 1. To set different values for these parameters, use the **service rfp local-endpoint** command with the **ccm-interval**, **level**, or **admin-state** parameters. For example:

```
-> service rfp 1 local-endpoint 10 ccm-interval interval10s level 6 admin-state
   disable type spb
```

In this example, RFP domain 1 with local MEP ID 10 is created with the CCM interval set to 10 seconds, the domain level set to 6, and the administrative status disabled.

The parameter values used to created an RFP domain are used to create a reserved Ethernet OAM domain on the local switch to which the RFP domain is mapped. The reserved OAM domain is given the name "RFP_OVER_SPB_DOMAIN_LEVELx", where x is the number specified with the **level** parameter. It is important to note that each RFP domain created must use a different level number. For example, if RFP domain 1 uses level 7, then RFP domain 2 must use a different level number (for example, level 6).

All the reserved OAM domains that are automatically created for RFP domains are assigned to the same "RFP_OVER_SPB_ASSOCIATION" Maintenance Association (MA).

Use the **show service rfp configuration** to display the Ethernet OAM domain parameters configured and associated with the RFP domain. For example, the following shows the OAM domain reserved for the RFP 1 domain ID:

```
-> show service rfp configuration
Total Number of RFP domains - 1

RFP Domain Number        : 1
Admin Status             : Enabled
Level                    : 7
Type                     : SPB
Maintenance Domain       : RFP_OVER_SPB_DOMAIN_LEVEL7
Maintenance Association  : RFP_OVER_SPB_ASSOCIATION
Control B-VLAN           : 500
Virtual UP MEP ID        : 10
CCM Interval             : 10 minutes
Remote Endpoint            Service Id
-----------------------+-------------
```

In this example, the "Remote Endpoint" and "Service Id" fields are blank because a list of remote end points and SPB services has not yet been created for RFP domain 1. A remote end point list provides a list of remote MEP IDs (local MEP IDs configured on other BEBs) and a list of SPB service IDs that are active on the local BEB. CCM packets are sent to all the remote MEP IDs on the list to advertise the status of the local SAP ports and I-SIDs associated with the specified SPB services.

To create a remote end point list for RFP domain 1, use the **service rfp remote-endpoint** command. For example:

```
-> service rfp 1 remote-endpoint 2 service-id 10-12
```

In this example, remote end point 2 is the MEP ID that identifies a remote BEB that is participating in the same RFP domain. The service IDs 10, 11, and 12 are the SPB services bound to the RFP 1 domain.

Creating a remote end point list triggers the transmission of CCM packets carrying I-SID and port status information related to the specified SPB services. Configure a different remote end point for each remote BEB that needs to receive the CCM packet information for a specific service. For example, the following commands add MEP ID 3 and 4 as remote end points to receive status for services 13 and 14:

```
-> service rfp 1 remote-endpoint 3 service-id 13
-> service rfp 1 remote-endpoint 4 service-id 14
```

Use the **show service rfp configuration** to display the RFP Ethernet OAM domain configuration showing the remote end points added to RFP domain 1. For example:

```
-> show service rfp configuration
Total Number of RFP domains - 1

RFP Domain Number        : 1
Admin Status             : Enabled
Level                    : 7
Type                     : SPB
Maintenance Domain       : RFP_OVER_SPB_DOMAIN_LEVEL7
Maintenance Association  : RFP_OVER_SPB_ASSOCIATION
Control B-VLAN           : 500
Virtual UP MEP ID        : 10
CCM Interval             : 10 minutes
```

```
Remote Endpoint          Service Id
------------------------+-------------
2                          10
2                          11
2                          12
3                          13
4                          14
```

Deleting the RFP Domain

To remove an RFP domain configuration from the switch, use the following command**:**

```
-> no service rfp 1
```

The above command removes all of the RFP configuration items, including the reserved Ethernet OAM domain for the specified RFP ID.

Modifying the local MEP ID

To change a local end point (MEP ID), first set the ID to zero using the **no** form of the **service rfp local-endpoint** command and specify the existing ID number. For example:

```
-> no service rfp 1 local-endpoint 10
```

Next, set the local MEP ID to a different value. For example, the following command sets a new value of 15 for the local MEP ID:

```
-> service rfp 1 local-endpoint 15
```

Removing an SPB Service from RFP Monitoring

To discontinue RFP monitoring of SPB service instances within an RFP domain, remove the local MEP ID from the remote end point list on each BEB that terminates the service. To remove a MEP ID from a remote end point list, use the **no** form of the **service rfp remote-endpoint** command. For example:

```
-> no service rfp 1 remote-endpoint 2
```

In this example, MEP ID 2 is removed from the remote end point list along with all SBP services associated with MEP ID 1. RFP will no longer monitor and advertise the status of local services to remote MEP ID 2.

To remove a specific SPB service from a remote end point list, use the **no** form of the **service rfp remote-endpoint** command with the **service-id** parameter. For example, the following command removes SPB service 10 associated with MEP ID 2:

```
-> no service rfp 1 remote-endpoint 2 service-id 10
```

In this example, SPB service 10 was removed from the end point list for MEP ID 2. However, RFP will continue to monitor and advertise all other services to this remote end point.

It is important to consider that when a MEP ID or a specific SPB service ID is removed from the end point list on the local switch, a port violation will occur. This can cause an undesirable service interruption when attempting to simply unbind a service from an RFP domain. To avoid a port violation condition, remove the SPB service ID from both ends of the SPB tunnel at the same time.

Verifying the RFP for SPB Configuration

To verify the connectivity between remote end points within an RFP domain, use the **show service rfp** command. For example:

```
-> show service rfp

Local system (Name : SystemId) =  Edge-39 : e8e7.326c.4a39
Total number of services information  = 2
Total number of RFP domain  = 3
RFP    Remote   RMEP      System                        B-VLAN ISID    Service Admin
       EndPoint Status    (Name : SystemId)                            Id     State
------+--------+--------+-------------------------+------+-------+-------+--------
1      2        RMEP_OK  Edge-43: 00:e0:b1:e7:09:a3 500    1001    10     Enabled
1      2        RMEP_OK  Edge-43: 00:e0:b1:e7:09:a3 500    1002    11     Enabled
1      2        RMEP_OK  Edge-43: 00:e0:b1:e7:09:a3 500    1003    12     Enabled
```

To verify the status of the local SAP associated with the RFP domain, use the **show service rfp** command with the **local-sap-status** parameter. For example:

```
-> show service rfp 1 local-sap-status

Local endpoint ID = 10
Local system (Name : SystemId)  = Edge-39 : e8e7.326c.4a39
Service  Sap               Admin        Oper          Remote        R-Endpoint
Id                                                     Endpoint      Status
--------+-----------------+------------+-------------+--------------+-----------
10       sap:1/12:all      Enabled      Down          2             RMEP_OK
11       sap:1/11:all      Enabled      Down          2             RMEP_OK
12       sap:1/10:all      Enabled      Down          2             RMEP_OK
```

As previously described, use the **show service rfp configuration** command to display the parameter values associated with the RFP domain. For example:

```
  -> show service rfp configuration
  Total Number of RFP domains - 1

  RFP Domain Number       : 1
  Admin Status            : Enabled
  Level                   : 7
  Type                    : SPB
  Maintenance Domain      : RFP_OVER_SPB_DOMAIN_LEVEL7
  Maintenance Association : RFP_OVER_SPB_ASSOCIATION
  Control B-VLAN          : 500
  Virtual UP MEP ID       : 10
  CCM Interval            : 10 minutes
  Remote Endpoint       Service Id
  ----------------------+-------------
  2                       10
  2                       11
  2                       12
```

RFP for SPB Configuration Example

This section contains CLI command examples used to configure the RFP domain functionality deployed in the following sample SPB network topology:
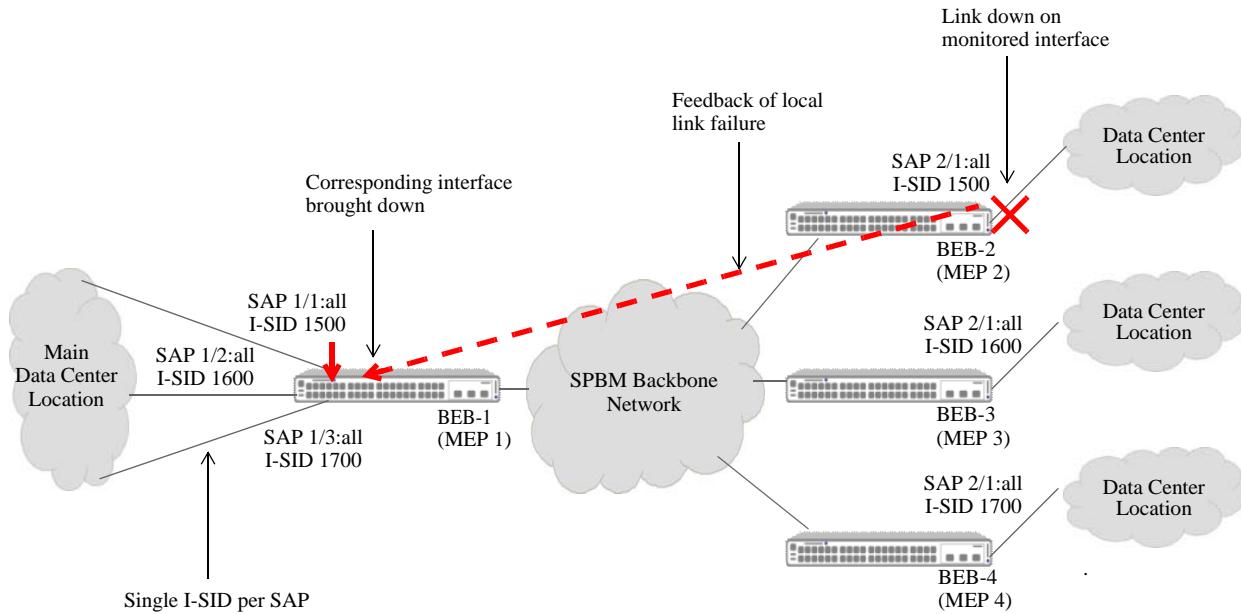

Figure 9: RFP for SPB Example

In this topology, RFP domain 1 is created by configuring a Maintenance End Point (MEP) on each OmniSwitch serving as an SPB BEB. The local MEP ID associated with each switch (MEP 1, 2, 3, and 4) identifies that switch as a participant in the RFP domain. As shown in the above diagram:

- Three I-SIDs (1500, 1600, and 1700) are each bound to a separate SAP port (1/1, 1/2, and 1/3) on BEB-1. This represents one end of each SPB service.

- The other end of each SPB service is bound to SAP ports on BEB-2 (I-SID 1500), BEB-3 (I-SID 1600), and BEB-3 (I-SID 1700).

- The RFP configuration on BEB-1 specifies the MEP ID of the other three BEBs as remote end points to which the local status of the three I-SIDS and SAP ports is advertised using CCM packets.

- The RFP configuration on BEB-2 specifies the MEP ID of BEB-1 as a remote end point to which the local status of I-SID 1500 and SAP port 2/1 is advertised using CCM packets.

- The RFP configuration on BEB-3 specifies the MEP ID of BEB-1 as a remote end point to which the local status of I-SID 1600 and SAP port 2/1 is advertised using CCM packets.

- The RFP configuration on BEB-4 specifies the MEP ID of BEB-1 as a remote end point to which the local status of I-SID 1700 and SAP port 2/1 is advertised using CCM packets.

- When SAP port 2/1 goes down on BEB-2, the port down status is reported in the CCM transmitted from BEB-2 to BEB-1.

- When BEB-1 receives the CCM packet from BEB-2 and detects the port down status, BEB-1 administratively downs the corresponding SAP port 1/1. The service associated with I-SID 1500 stops on both ends of the service (BEB-1 and BEB-2).

- When the downed port on BEB-2 is brought back up, BEB-1 receives the port up status from BEB-2 and brings the local SAP port 1/1 back up as well.

The following CLI command examples include the SPB commands used to create the SPB service layer that RFP will monitor.

BEB-1:

```
-> spb bvlan 4001 admin-state enable
-> spb isis bvlan 4001 ect-id 2
-> spb isis control-bvlan 4001
-> spb isis interface port 1/20
-> spb isis admin-state enable
-> service access port 1/1
-> service access port 1/2
-> service access port 1/3
-> service spb 1 isid 1500 bvlan 4001 admin-state enable
-> service spb 2 isid 1600 bvlan 4001 admin-state enable
-> service spb 3 isid 1700 bvlan 4001 admin-state enable
-> service spb 1 sap port 1/1:all admin-state enable
-> service spb 2 sap port 1/2:all admin-state enable
-> service spb 3 sap port 1/3:all admin-state enable

-> service rfp 1 local-endpoint 1 ccm-interval interval100ms type spb
-> service rfp 1 remote-endpoint 2 service-id 1 admin-status enable
-> service rfp 1 remote-endpoint 3 service-id 2 admin-status enable
-> service rfp 1 remote-endpoint 4 service-id 3 admin-status enable
```

BEB-2:

```
-> spb bvlan 4001 admin-state enable
-> spb isis bvlan 4001 ect-id 2
-> spb isis control-bvlan 4001
-> spb isis interface port 1/21
-> spb isis admin-state enable
-> service access port 2/1
-> service spb 1 isid 1500 bvlan 4001 admin-state enable
-> service spb 1 sap port 2/1:all admin-state enable

-> service rfp 1 local-endpoint 2 ccm-interval interval100ms type spb
-> service rfp 1 remote-endpoint 1 service-id 1 admin-status enable
```

BEB-3:

```
-> spb bvlan 4001 admin-state enable
-> spb isis bvlan 4001 ect-id 2
-> spb isis control-bvlan 4001
-> spb isis interface port 1/22
-> spb isis admin-state enable
-> service access port 2/1
-> service spb 2 isid 1600 bvlan 4001 admin-state enable
-> service spb 2 sap port 2/1:all admin-state enable

-> service rfp 1 local-endpoint 3 ccm-interval interval100ms type spb
-> service rfp 1 remote-endpoint 1 service-id 2 admin-status enable
```

BEB-4:

```
-> spb bvlan 4001 admin-state enable
-> spb isis bvlan 4001 ect-id 2
-> spb isis control-bvlan 4001
-> spb isis interface port 1/23
-> spb isis admin-state enable
-> service access port 2/1
-> service spb 3 isid 1700 bvlan 4001 admin-state enable
-> service spb 3 sap port 2/1:all admin-state enable

-> service rfp 1 local-endpoint 4 ccm-interval interval100ms type spb
-> service rfp 1 remote-endpoint 1 service-id 3 admin-status enable
```

Configuring IP over SPB

As described in section "IP over SPBM" on page 3-17, there are two approaches for routing L3 traffic over a L2 SPBM backbone network: VPN-Lite and L3 VPN (SPB-ISIS). The following general steps are used to configure each one of these solutions:

1 Configure an L3 VPN interface. Identify the BEBs that will participate in routing L3 traffic through the SPBM core. On each of these BEBs, configure the required L3 VPN interface configuration. There are two options for configuring this type of interface based on the switch platform: an external loopback port configuration (OmniSwitch 6900, 6860, 6865) or a service-based IP interface (OmniSwitch 9900).

a. Configure an L3 VPN loopback interface (OmniSwitch 6900, 6860, 6865). The following commands create an external loopback port configuration that will serve as an L3 VPN interface:

```
-> vlan 200
-> vlan 200 members 1/1 tagged
-> spb bvlan 500
-> service access port 1/2
-> service 1000 spb isid 1000 bvlan 500 admin-state enable
-> service 1000 sap port 1/2:200
-> vrf create vrf-1
vrf-1::-> vrf-1 ip interface L3vpn1 vlan 200 address 10.1.1.1/24
```

Once the above loopback port configuration is defined, use a physical cable to connect port 1/1 to access port 1/2. The SPB BVLAN and I-SID are required to create the SAP for access port 1/2.

b. Configure an L3 VPN service-based interface (OmniSwitch 9900). The following commands provide an example for creating an L3 VPN service-based interface:

```
-> vrf create vrf-1
vrf-1::-> ip interface L3vpn2 service 10 address 20.1.1.1/24
```

In this example, the "L3vpn2" interface is created in the "vrf-1" instance and is bound to an SPB service.

2 Configure VPN-Lite routing protocols. If using the VPN-Lite approach, configure the routing protocols on the IP interfaces created in Step 1. Otherwise, skip this step (Step 2) and go to Step 3. For example:

```
vrf-1::-> ip static-route 20.0.0.0/24 gateway 10.1.1.1
vrf-1::-> ip static-route 10.0.0.0/24 gateway 20.1.1.1
```

The VPN-Lite approach requires configuring routing protocols on the L3 VPN interface to exchange routes. The remaining steps in this section are used to configure the L3 VPN approach, which uses ISIS-SPB to exchange routes.

3 Configure VRF-ISID bindings for the L3 VPN (SPB-ISIS). A VRF-ISID binding identifies the L3 VPN interface that will route L3 traffic between the VLAN domain (VRF instance) and the SPB service domain (I-SID). VRF import and export commands are used to exchange routes between the VRF and the I-SID specified in the binding configuration. For example:

```
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 10.1.1.1 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
```

In this example, "vrf-1" is bound to SPB I-SID 1000 and gateway 10.1.1.1 identifies the L3 VPN interface. All routes in "vrf-1" are exported to the Global Route Manager (GRM), which then exports

---

the routes to I-SID 1000. The last command in this sequence sets up the import of I-SID 1000 routes from the GRM into "vrf-1". The **all-routes** parameter specifies that no route-map filtering is applied to exported or imported routes; all routes are allowed.

4 Optionally configure route redistribution between VRFs and/or I-SIDs. Route redistribution is configurable between a VRF and I-SID or between two I-SIDs (inter-I-SID route leaking). For example:

```
-> spb ipvpn redist source-isid 2000 destination-isid 1000 all-routes
-> spb ipvpn redist source-vrf-2 destination-isid 2000 all-routes
```

Verifying L3 VPN Configuration and Routes

VRFs are bound to I-SIDs to identify a VRF mapping to a specific SPB service instance for the purposes of exchanging routes between the VRF and I-SID via the switch GRM. To verify the VRF mapping configuration on the local switch, use the **show spb ipvpn bind** command. For example:

```
-> show spb ipvpn bind
Legend: * indicates bind entry is active
SPB IPVPN Bind Table:
      VRF              ISID        Gateway             Route-Map
--------------------+------------+---------------------+----------------------
* vrf-1               1000        1.1.1.2
* vrf-2               2000        2.2.2.2

Total Bind Entries: 2
```

In addition to exchanging routes between VRFs and I-SIDs, it is also possible to configure redistribution of routes between two I-SIDs or between a VRF and an I-SID. To verify the redistribution configuration for L3 VPN routes, use the **show spb ipvpn redist** command. For example:

```
-> show spb ipvpn redist
Legend: * indicates redist entry is active
SPB IPVPN Redist ISID Table:
  Source-ISID         Destination-ISID    Route-Map
---------------------+-------------------+-------------------
* 4001                4003
* 4003                4001

Total Redist ISID Entries: 2

Legend: * indicates redist entry is active
SPB IPVPN Redist VRF Table:
  Source-VRF          Destination-ISID    Route-Map
---------------------+-------------------+-------------------

Total Redist Vrf Entries: 0
```

To display the L3 VPN route table, use the **show spb ipvpn route-table**. For example:

```
-> show spb ipvpn route-table
Legend: * indicates IPVPN route has matching locally configured ISID
SPB IPVPN Route Table:
                                        Source Bridge
     ISID  Destination       Gateway        (Name : BMAC)              Metric
---------+-------------------+----------------+---------------------------+------
*   4001  1.1.1.0/24          1.1.1.1        L2-DUT1 : 00:e0:b1:db:c3:65    1
*   4001  1.1.1.0/24          1.1.1.2        L2-DEV1 : e8:e7:32:00:23:f9    1
*   4001  2.2.2.0/24          1.1.1.2        L2-DEV1 : e8:e7:32:00:23:f9    1
```

```
*    4001  10.10.10.0/24       1.1.1.1           L2-DUT1 : 00:e0:b1:db:c3:65    1
*    4001  15.1.1.0/24         1.1.1.1           L2-DUT1 : 00:e0:b1:db:c3:65    1
*    4003  1.1.1.0/24          2.2.2.2           L2-DEV1 : e8:e7:32:00:23:f9    1
*    4003  2.2.2.0/24          2.2.2.1           L2-DUT2 : 00:e0:b1:dd:99:db    1
*    4003  2.2.2.0/24          2.2.2.2           L2-DEV1 : e8:e7:32:00:23:f9    1
*    4003  10.10.10.0/24       2.2.2.2           L2-DEV1 : e8:e7:32:00:23:f9    1
*    4003  15.1.1.0/24         2.2.2.2           L2-DEV1 : e8:e7:32:00:23:f9    1

-> show spb ipvpn route-table isid 4003
Legend: * indicates IPVPN route has matching locally configured ISID
SPB IPVPN Route Table:
                                               Source Bridge
     ISID  Destination         Gateway         (Name : BMAC)              Metric
---------+-------------------+----------------+--------------------------+------
*    4003  1.1.1.0/24          2.2.2.2           L2-DEV1 : e8:e7:32:00:23:f9    1
*    4003  2.2.2.0/24          2.2.2.1           L2-DUT2 : 00:e0:b1:dd:99:db    1
*    4003  2.2.2.0/24          2.2.2.2           L2-DEV1 : e8:e7:32:00:23:f9    1
*    4003  10.10.10.0/24       2.2.2.2           L2-DEV1 : e8:e7:32:00:23:f9    1
*    4003  15.1.1.0/24         2.2.2.2           L2-DEV1 : e8:e7:32:00:23:f9    1
```

IP over SPB Configuration Examples

This section contains diagrams and CLI command examples for configuring the following IP over SPB scenarios:

-

-

-

-

L3 VPN Loopback and Service-based Interfaces: Two I-SIDs, One VRF

In this sample IP over SPB topology, Network A can communicate with Network B across two I-SIDs. This scenario could be expanded to connect multiple customer sites together to form a VPN cloud.
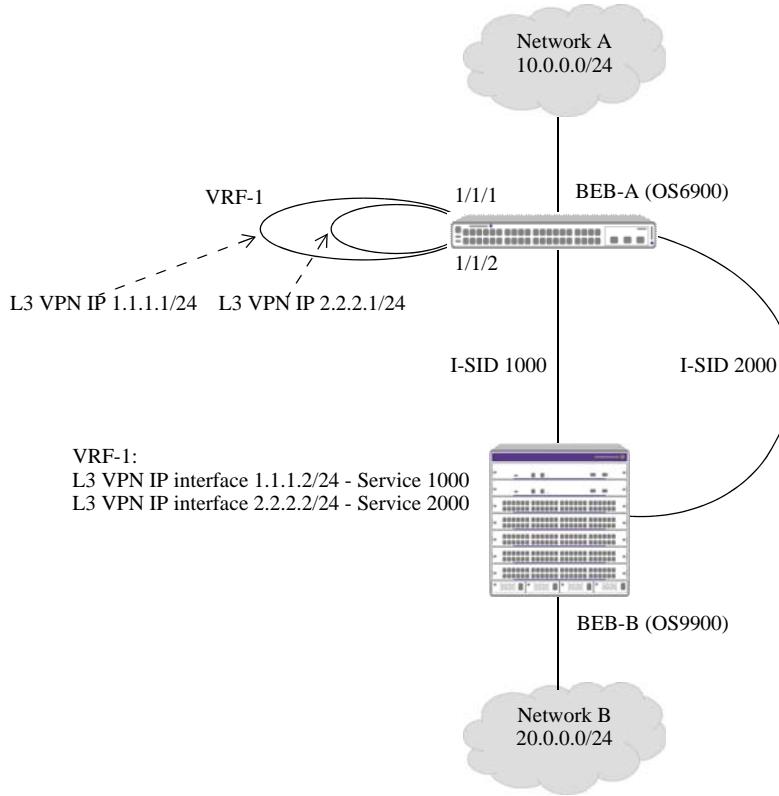
Figure 10: L3 VPN Loopback and Service-based Interfaces

In this topology,

- A physical loopback port configuration is required on BEB-A (OS6900). Port 1/1/1 is the L3 VPN router port, port 1/1/2 is the L3 VPN access port, and VLAN 200 and VLAN 400 are the L3 VPN VLANs.

  – Port 1/1/1 is tagged with VLANs 200 and 400.

  – Access port 1/1/2 is assigned to SAPs that are each associated with an I-SID (an SPB service instance) that will forward VLAN 200 and 400 traffic through the SPB backbone network.

  – A physical cable is connected to port 1/1/1 and to port 1/1/2 to create the connection between the VLAN and service domains.

  – An L3 VPN interface is configured on VLAN 200 (1.1.1.1/24) and VLAN 400 (2.2.2.1/24).

- A physical loopback port configuration is not used on BEB-B (OS9900). Instead, an L3 VPN interface is configured on service 1000 (1.1.1.2/24) and service 2000 (2.2.2.2/24).

- On BEB-A and BEB-B, the VPN-Lite or L3 VPN solution is configured.

The following CLI command examples are used to configure the sample IP over SPB topology shown in "Figure 10: L3 VPN Loopback and Service-based Interfaces" on page 3-59.

BEB-A:
```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 vlan 200 address 1.1.1.1/24
vrf-1::-> ip interface l3vpn2 vlan 400 address 2.2.2.1/24
```

BEB-B (OS9900):
```
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 service 1000 address 1.1.1.2/24
vrf-1::-> ip interface l3vpn2 service 2000 address 2.2.2.2/24
```

VPN-Lite

The following commands are used only in a VPN-Lite configuration:

BEB-A:
```
vrf-1::-> ip static-route 20.0.0.0/24 gateway 1.1.1.2
vrf-1::-> ip static-route 20.0.0.0/24 gateway 2.2.2.2
```

BEB-B (OS9900):
```
vrf-1::-> ip static-route 10.0.0.0/24 gateway 1.1.1.1
vrf-1::-> ip static-route 10.0.0.0/24 gateway 2.2.2.1
```

L3 VPN

The following commands are used only in an L3 VPN configuration:

BEB-A:
```
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.1 all-routes
-> spb ipvpn bind vrf-1 isid 2000 gateway 2.2.2.1 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
vrf-1::-> ip import isid 2000 all-routes
```

BEB-B (OS9900):
```
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.2 all-routes
-> spb ipvpn bind vrf-1 isid 2000 gateway 2.2.2.2 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
vrf-1::-> ip import isid 2000 all-routes
```

L3 VPN Service-based Interfaces: Two I-SIDS, Two VRFs

In this sample IP over SPB topology, Customer Edge (CE) devices can communicate with other CE devices through L3 VPN services that traverse the SPB backbone network. Because the OmniSwitch 9900 is used in this example topology, the L3 VPN interfaces are defined by creating an IP interface within each VRF instance that is bound to an SPB service.
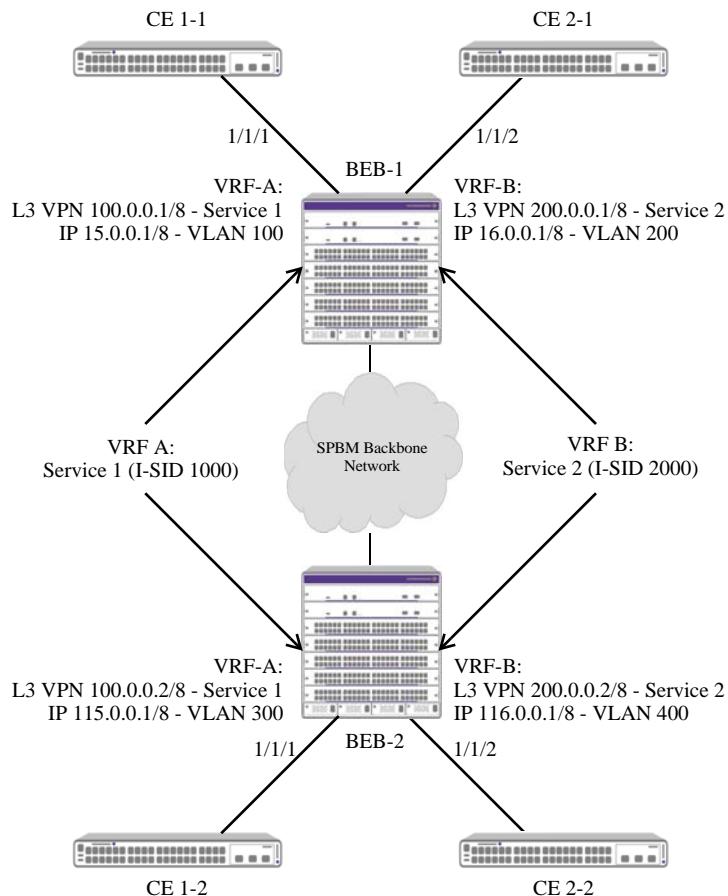

Figure 11: L3 VPN Service-based Interfaces

In this topology, Customer 1 requires L3 VPN services between CE 1-1 and CE 1-2, and Customer 2 requires L3 VPN services between CD 2-1 and CE 2-2.

- BEB-1 is connected to two customer edge devices (CE 1-1 and CE 2-1).

- BEB-2 is connected to two customer edge devices (CE 1-2 and CE 2-2.

- SPB service 1/I-SID 1000 and VRF A are dedicated to Customer 1.

- SPB service 2/I-SID 2000 and VRF B are dedicated to Customer 2.

- An IP interface is configured in VRF A and bound to SPB service 1 to define an L3 VPN interface for Customer 1.

- An IP interface is configured in VRF B and bound to SPB service 2 to define an L3 VPN interface for Customer 2.

- VPN routes are learned on the BEB switches through L3 VPN (SPB-ISIS) or VPN-lite and are then installed into the appropriate VRFs.

- The nexthop interface for VPN routes is through the L3 VPN interfaces. For example,
  - The VPN routes in BEB-1 for VRF-A are installed with the next hop set to100.0.0.2, and the VPN routes in BEB-2 for VRF-A, are installed with the nexthop set to100.0.0.1.
  - The VPN routes in BEB-1 for VRF-B are installed with the next hop set to 200.0.0.2, and the VPN routes in BEB-2 for VRF-B are installed with the next hop set to 200.0.0.1.

The following CLI command examples are used to configure the sample IP over SPB topology shown in .

BEB-1:
```
-> vlan 100
-> vlan 100 members port 1/1/1 tagged
-> vrf create vrf-A
vrf-A::-> ip interface ip-a1 address 15.0.0.1/8 vlan 100
vrf-A::-> ip interface l3vpn address 100.0.0.1/8 service 1
vrf-A::-> vrf default

-> vlan 200
-> vlan 200 members port 1/1/2 tagged
-> vrf create vrf-B
vrf-B::-> ip interface ip-b1 address 16.0.0.1/8 vlan 200
vrf-B::-> ip interface l3vpn address 200.0.0.1/8 service 2
```

BEB-2:
```
-> vlan 300
-> vlan 300 members port 1/1/1 tagged
-> vrf create vrf-A
vrf-A::-> ip interface ip-a1 address 115.0.0.1/8 vlan 300
vrf-A::-> ip interface l3vpn address 100.0.0.2/8 service 1

-> vlan 400
-> vlan 400 members port 1/1/2 tagged
-> vrf create vrf-B
vrf-B::-> ip interface ip-b1 address 116.0.0.1/8 vlan 400
vrf-B::-> ip interface l3vpn address 200.0.0.2/8 service 2
```

VPN-Lite

The following commands are used only when configuring a VPN-Lite configuration.

BEB-1:
```
vrf-A::-> ip static-route 115.0.0.0/8 gateway 100.0.0.2
vrf-B::-> ip static-route 116.0.0.0/8 gateway 200.0.0.2
```

BEB-2:
```
vrf-A::-> ip static-route 15.0.0.0/8 gateway 100.0.0.1
vrf-B::-> ip static-route 16.0.0.0/8 gateway 200.0.0.1
```

L3 VPN

The following commands are used only when configuring an L3 VPN (SPB-ISIS) configuration.

BEB-1:
```
vrf-A::-> ip export all-routes
vrf-A::-> vrf default
```

```
-> spb ipvpn bind vrf-A isid 1000 gateway 100.0.0.1 all-routes
-> vrf vrf-A
vrf-A::-> ip import isid 1000 all-routes
vrf A::-> vrf default

-> vrf vrf-B
vrf-B::-> ip export all-routes
vrf-B::-> vrf default
-> spb ipvpn bind vrf-B isid 2000 gateway 200.0.0.1 all-routes
-> vrf vrf-B
vrf-B::-> ip import isid 2000 all-routes
vrf B::-> vrf default
```

BEB-2:

```
vrf-A::-> ip export all-routes
vrf-A::-> vrf default
-> spb ipvpn bind vrf-A isid 1000 gateway 100.0.0.2 all-routes
-> vrf vrf-A
vrf-A::-> ip import isid 1000 all-routes
vrf A::-> vrf default

-> vrf vrf-B
vrf-B::-> ip export all-routes
vrf-B::-> vrf default
-> spb ipvpn bind vrf-B isid 2000 gateway 200.0.0.2 all-routes
-> vrf vrf-B
vrf-B::-> ip import isid 2000 all-routes
vrf B::-> vrf default
```

L3 VPN Loopback Interfaces: I-SID Routing in One VRF

In this sample IP over SPB configuration, Networks A and B can communicate with each other within the same VRF but on different I-SIDs due to the routing (redistribution) between I-SID 1000 and 2000 on BEB-C. In addition, Network C is also able to communicate with Networks A and B.
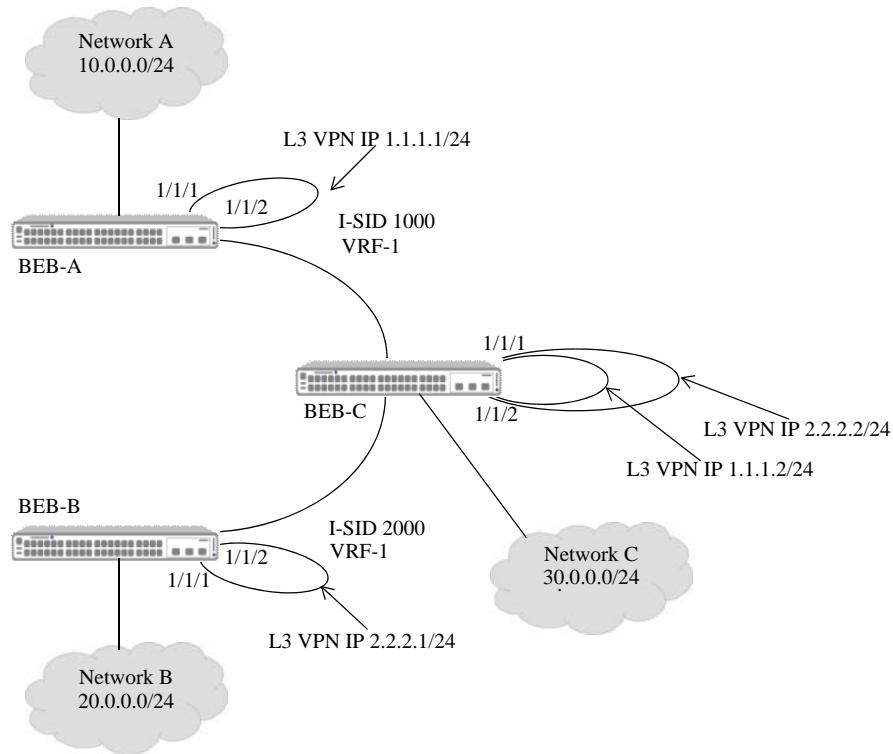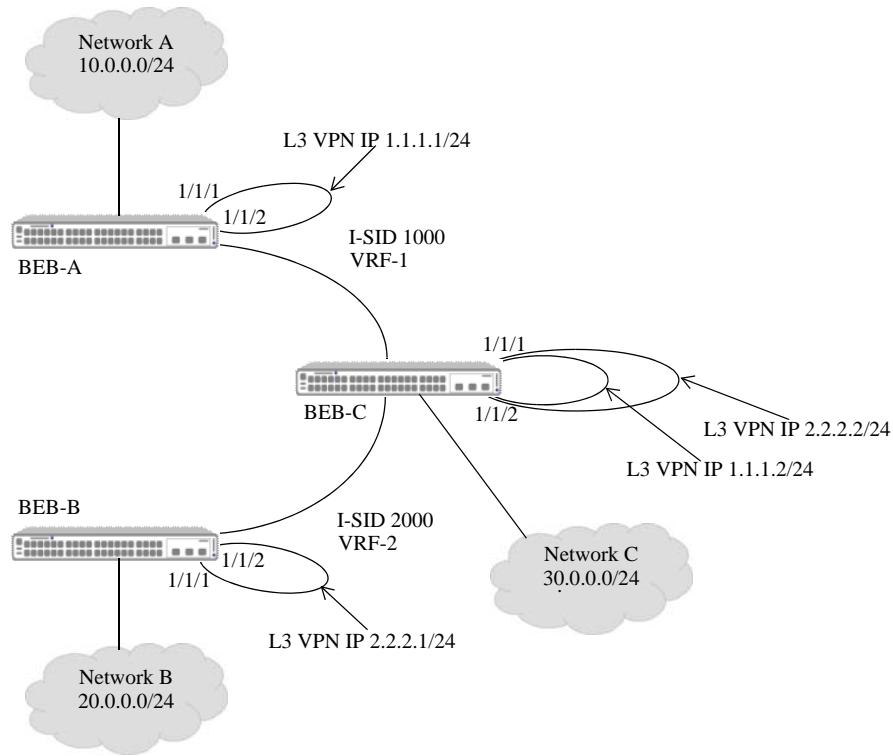

Figure 12: Inter-ISID Routing Example (One VRF)

In this topology, Network A binds to I-SID 1000 in VRF-1, Network B binds to I-SID 2000 in VRF-1, and Network C binds to both I-SIDs in VRF-1. The following CLI command examples are used to configure the sample IP over SPB topology shown in "Figure 12: Inter-ISID Routing Example (One VRF)" on page 3-64.

BEB-A:

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 vlan 200 address 1.1.1.1/24
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.1 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
```

BEB-B

```
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
```

```
-> service access port 1/1/2
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn2 vlan 400 address 2.2.2.1/24
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 2000 gateway 2.2.2.1 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 2000 all-routes
```

BEB-C:

```
-> vlan 200
-> vlan 200 members port 1/1 tagged
-> vlan 400
-> vlan 400 members port 1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 vlan 200 address 1.1.1.2/24
vrf-1::-> ip interface l3vpn2 vlan 400 address 2.2.2.2/24
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.2 all-routes
-> spb ipvpn bind vrf-1 isid 2000 gateway 2.2.2.2 all-routes
-> spb ipvpn redist source-isid 1000 destination-isid 2000 all-routes
-> spb ipvpn redist source-isid 2000 destination-isid 1000 all-routes
-> vrf vrf-1
vrf-1::-> ip import all-routes
vrf-1::-> ip export all-routes
```

L3 VPN Loopback Interfaces: I-SID Routing in Two VRFs

In this sample IP over SPB configuration, Networks A and B can communicate with each other between two different VRFs on different I-SIDs due to the routing (redistribution) between I-SID 1000 and 2000 on BEB-C. In addition, Network C is also able to communicate with Networks A and B.

Figure 13: Inter-ISID Routing Example (Two VRFs)

In this topology, Network A binds to I-SID 1000 in VRF-1, Network B binds to I-SID 2000 in VRF-2, and Network C binds to both I-SIDs in VRF-1. The following CLI command examples are used to configure the sample IP over SPB topology shown in .

BEB-A

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 vlan 200 address 1.1.1.1/24
vrf-1::-> ip export all-routes
vrf-1::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.1 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
```

BEB-B

```
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
```

```
-> service access port 1/1/2
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-2
vrf-2::-> ip interface l3vpn2 vlan 400 address 2.2.2.1/24
vrf-2::-> ip export all-routes
vrf-2::-> vrf default
-> spb ipvpn bind vrf-1 isid 2000 gateway 2.2.2.1 all-routes
-> vrf vrf-2
vrf-2::-> ip import isid 2000 all-routes
```

BEB-C

```
-> vlan 200
-> vlan 200 members port 1/1/1 tagged
-> vlan 400
-> vlan 400 members port 1/1/1 tagged
-> service access port 1/1/2
-> service 1000 spb isid 1000 bvlan 40 admin-state enable
-> service 1000 sap port 1/1/2:200 admin-state enable
-> service 2000 spb isid 2000 bvlan 41 admin-state enable
-> service 2000 sap port 1/1/2:400 admin-state enable
-> vrf create vrf-1
vrf-1::-> ip interface l3vpn1 vlan 200 address 1.1.1.2/24
vrf-1::-> vrf create vrf-2
vrf-2::-> ip interface l3vpn2 vlan 400 address 2.2.2.2/24
vrf-2::-> vrf default
-> spb ipvpn bind vrf-1 isid 1000 gateway 1.1.1.2 all-routes
-> spb ipvpn bind vrf-2 isid 2000 gateway 2.2.2.2 all-routes
-> spb ipvpn redist source-isid 1000 destination-isid 2000 all-routes
-> spb ipvpn redist source-isid 2000 destination-isid 1000 all-routes
-> spb ipvpn redist source-vrf vrf-1 destination-isid 2000 all-routes
-> vrf vrf-1
vrf-1::-> ip import isid 1000 all-routes
vrf-1::-> ip import isid 2000 all-routes
vrf-1::-> ip export all-routes
vrf-1::-> vrf vrf-2
vrf-2::-> ip import vrf-1 all-routes
vrf-2::-> ip import isid 1000 all-routes
vrf-2::-> ip import isid 2000 all-routes
```

Verifying the SPB Backbone and Services

Displaying the SPBM configuration is helpful to verify the actual configuration on each SPB switch in the topology and to troubleshoot ISIS-SPB backbone and SPB service connectivity.

Verifying the ISIS-SPB Backbone Configuration

To display information about the ISIS-SPB infrastructure (backbone), use the **show** commands listed in this section.

show spb isis info	Displays the global status and configuration for the ISIS-SPB instance on the switch.
show spb isis bvlans	Displays the backbone VLAN (BVLAN) configuration for the switch.
show spb isis interface	Displays the SPB interface (network port) configuration for the switch.
show spb isis adjacency	Displays information about the ISIS-SPB adjacencies created for the SPB switch.
show spb isis database	Displays ISIS-SPB topology information maintained in the link state database (LSDB).
show spb isis nodes	Displays the discovered node-level parameter values for all of the ISIS-SPB switches participating in the topology.
show spb isis unicast-table	Displays the unicast forwarding information for the BVLAN topology.
show spb isis services	Displays a network-wide view of existing services to help verify that SPB services are correctly advertised and learned by ISIS-SPB
show spb isis spf	Displays the shortest path first (SPF) information to all known SPB switches for a specific BVLAN.
show spb isis multicast-table	Displays the multicast forwarding entries for services.
show spb isis multicast-sources	Displays all the known multicast sources across the SPB domain and BVLANs.
show spb isis multicast-sources-spf	Displays the shortest path first (SPF) readability for a known multicast source bridge for a specific BVLAN.
show spb isis ingress-mac-filter	Displays the ingress MAC filter for multicast traffic for a given BVLAN operating in the (*,G) mode.

Verifying the SPB Service Configuration

To display information about the Service Manager configuration for SPB service connectivity, use the **show** commands listed in this section

show service access   Displays the service access (customer-facing) port configuration.

show service l2profile   Displays the Layer 2 profile definitions. These profiles are applied to service access ports to determine how Layer 2 control protocol frames are processed on these ports.

show service   Displays the service configuration.

show service ports   Displays all the virtual ports (SAPs, SDPs) that are associated with an SPB service.

show service spb sap   Displays the configuration information for the specified SAP ID associated with the specified service.

show service sdp   Displays the dynamic Service Distribution Point (SDP) configuration.

show service bind-sdp   Displays the dynamic SDP-to-service binding configuration.

show service debug-info   Displays debug information for the virtual ports associated with the SPB service.

show service info   Displays the global Service Manager configuration for the switch.

show service counters   Displays the traffic statistics for the specified SPB service and associated virtual ports.

clear service counters   Clears the traffic statistics for the specified SPB service and associated virtual ports.

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

4 Configuring a VXLAN Gateway

A Virtual eXtensible Local Area Network (VXLAN) is a Layer 2 overlay network that is used to segment and tunnel device traffic through a data center or cloud network infrastructure. The OmniSwitch implementation of this feature introduces the following benefits into the network:

- Provides Layer 2 connectivity between devices in the same VLAN over an IP transport network. For example, a Virtual Machine (VM) can communicate across a Layer 3 network with a remote VM as long as both VMs reside in the same VLAN domain on either side of the Layer 3 network.

- Increases the scalability of the network beyond the limit of 4096 VLANs. A VXLAN Network Identifier (VNI) is used to isolate VLAN traffic into logical network segments. Up to 16 million logical networks (segment IDs) are possible when VXLAN is implemented.

- Conserves MAC address table space by allowing duplicate MAC addresses to reside within the VXLAN domain, as long as each address is associated with a different VNI.

- Transparently extends the Layer 2 network by connecting VLANs from multiple hosts through VXLAN (UDP) tunnels.

- Provides Layer 2 migration of a Virtual Machine (VM) across a Layer 3 infrastructure to a remote server host; without VXLAN, Layer 2 migration is restricted to other servers within the local Layer 2

- Allows Layer 2 VM migration across a Layer 3 infrastructure to a remote server host; without VXLAN, VM migration occurs only within a Layer 2 infrastructure to a local server host.

- Provides a gateway device that sits at the edge of a VXLAN UDP tunnel to serve as a VXLAN Tunnel End Point (VTEP). An OmniSwitch VXLAN gateway offers a high bandwidth, low latency option for connecting VM traffic to remote servers or other VMs over a Layer 3 network.

In This Chapter

This chapter provides an overview about how the VXLAN feature works and how to configure VXLAN components through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

This chapter includes the following topics:

The acronyms and abbreviations used in this chapter are defined here:

VM	Virtual Machine
VNI	VXLAN Network Identifier (also referred to as a Segment ID)
VTEP	VXLAN Tunnel End Point
VXLAN	Virtual eXtensible Local Area Network
VXLAN Segment	A VXLAN Layer 2 overlay network.
VXLAN Gateway	A device on which traffic is forwarded between VLAN and VXLAN domains.
VXLAN Service	A service instance associated with a VNI.
PIM	Protocol Independent Multicast
BIDIR-PIM	Bidirectional PIM
SAP	Service Access Point
SDP	Service Distribution Point
SDP Bind	The binding of a VXLAN service to an SDP.

VXLAN Service Defaults

By default, there are no VXLAN service components configured for the switch. However, when a service is created, the following default values apply:

Parameter Description	Command	Default
VXLAN service administrative status.	**service admin-state**	Enable
VXLAN service multicast replication mode.	**service multicast-mode**	Hybrid
VXLAN service VLAN translation.	**service vlan-xlation**	Disabled
VXLAN service maximum transmission unit (MTU) value.	Not configurable at this time.	9194
VXLAN service statistics collection.	**service stats**	Disabled
VXLAN service description.	**service description**	None
Default profile automatically applied to access ports.	**service l2profile**	def-access-profile
Layer 2 profile that specifies how control packets are processed on service access ports.	**service access l2profile**	def-access-profile: STP, GVRP, MVRP = tunnel 802.3ad = peer 802.1x, 802.1ab, AMAP = drop
VLAN translation for the service access port.	**service access vlan-xlation**	Disabled
Service access point (SAP) administrative status.	**service sap admin-state**	Disabled
SAP encapsulation.	**service sap**	0 (untagged traffic)
SAP trust mode.	**service sap trusted**	Trusted
SAP statistics collection.	**service sap stats**	Disabled
SAP description.	**service sap description**	None

Quick Steps for Configuring a VXLAN Gateway

The implementation of the VXLAN feature is achieved through the service-based architecture that is available on the OmniSwitch. This section provides a brief tutorial for configuring an OmniSwitch to serve as a VXLAN gateway device.

1 Use the **service access** command to configure a port or link aggregate as a service access port on which Virtual Machine (VM) traffic is received.

```
-> service access port 1/1
-> service access port 2/1
```

2 Use the **service sdp vxlan** command to create a VXLAN tunnel interface that will handle the processing of VM frames and VXLAN encapsulated packets.

```
-> service sdp 100 vxlan multicast-group 224.2.1.1 ttl 20 description "PIM Group
224.2.1.1"
-> service sdp 200 vxlan far-end 10.10.10.2 description "Unicast to NodeB"
```

3 Use the **service vxlan** command to create a VXLAN service and associate that service with a VXLAN Network ID.

```
-> service 10 vxlan vnid 1000 multicast-mode tandem admin-state enable
-> service 20 vxlan vnid 2000 multicast-mode head-end admin-state enable
```

4 Use the **service sap** command to create a Service Access Point (SAP) by associating a VXLAN service with SAP ID. A SAP ID is comprised of a port or link aggregate and an encapsulation value that identifies the device traffic to associate with the service.

```
-> service 10 sap port 1/1:10 admin-state enable
-> service 20 sap port 2/1:all admin-state enable
```

5 Use the **service bind-sdp** command to bind the VXLAN services created in Step 3 to the SDPs created in Step 2.

```
-> service 10 bind-sdp 100 description "Bind to PIM Group 224.2.1.1"
-> service 20 bind-sdp 200 description "Unicast Bind to 10.10.10.2 VTEP"
```

6 Use the **ip interface** command to configure the Loopback0 interface that will serve as the source IP address of the VXLAN gateway (VXLAN Tunnel End Point).

```
-> ip interface Loopback0 address 11.255.14.102
```

Quick Steps for Bidirectional PIM

When a VXLAN tunnel interface (SDP on the OmniSwitch) is configured to use the Tandem multicast mode, PIM multicast routing is required to discover neighbor nodes and assign membership to VXLAN nodes that desire to be in the same multicast group. Using Bidirectional PIM (BIDIR-PIM) to provide this infrastructure is recommended. The following is a sample BIDIR-PIM configuration:

```
-> ip load pim
-> ip pim interface "v4lan100"
-> ip pim interface "v4lan200"
-> ip pim interface "Loopback0"
-> ip pim candidate-rp 10.0.0.2 224.0.0.0/4 bidir
-> ip pim cbsr 10.0.0.2
-> ip pim sparse admin-state enable
-> ip pim bidir admin-state enable
```

VXLAN Overview

The VXLAN feature is similar to other tunneling and network virtualization solutions, such as Shortest Path Bridging (SPB), in that an encapsulation technique is used to tunnel device traffic through the network. The technique implemented with the VXLAN feature encapsulates an Ethernet MAC frame received from a Virtual Machine (VM) into an IP packet with a UDP header, then forwards the packet on a Layer 3 network.

Configuring VXLAN components on the OmniSwitch allows the switch to operate as a VXLAN gateway device. This type of device connects VXLAN and non-VXLAN (traditional VLAN) segments. The following terms and definitions describe the VXLAN components discussed in this chapter.

- **VXLAN segment:** A VXLAN Layer 2 overlay network used to tunnel traffic between Virtual Machines (VMs). Each segment is identified with a VXLAN network identifier, which is similar to a VLAN ID (used to segment network traffic into virtual bridging domains). Only VMs associated with the same VXLAN segment can communicate with each other.

- **VXLAN network identifier (VNI):** A 24-bit number that identifies a VXLAN segment (also referred to as a VXLAN segment ID). A VNI is used to associate a VM MAC frame with a VXLAN segment when the frame is encapsulated with a VXLAN header.

- **VXLAN Tunnel Interface (VTI):** A UDP tunnel that forwards encapsulated VXLAN packets between VXLAN Tunnel End Points. On the OmniSwitch, the VTI function is provided through a Service Distribution Point (SDP) and the binding of VXLAN services to the SDP. A VTI is treated as just another Layer 2 interface within a bridging domain.

- **VXLAN Tunnel End Point (VTEP):** A switch configured with one or more VTIs. The VTEP provides an initiation and termination point for each VXLAN segment bound to the VTI. This is the point at which Layer 2 VM frames are encapsulated and sent through the tunnel or the encapsulation header is removed from a VXLAN packet and the Layer 2 VM frames are forwarded on a traditional VLAN domain.

- **VXLAN gateway**: A device that serves as a VTEP to transparently bridge traffic between VXLAN and traditional VLAN domains. A VXLAN gateway switch represents a single VTEP on which multiple VTIs may exist. On the OmniSwitch, a VTEP is identified by the IP address configured for the Loopback0 interface.

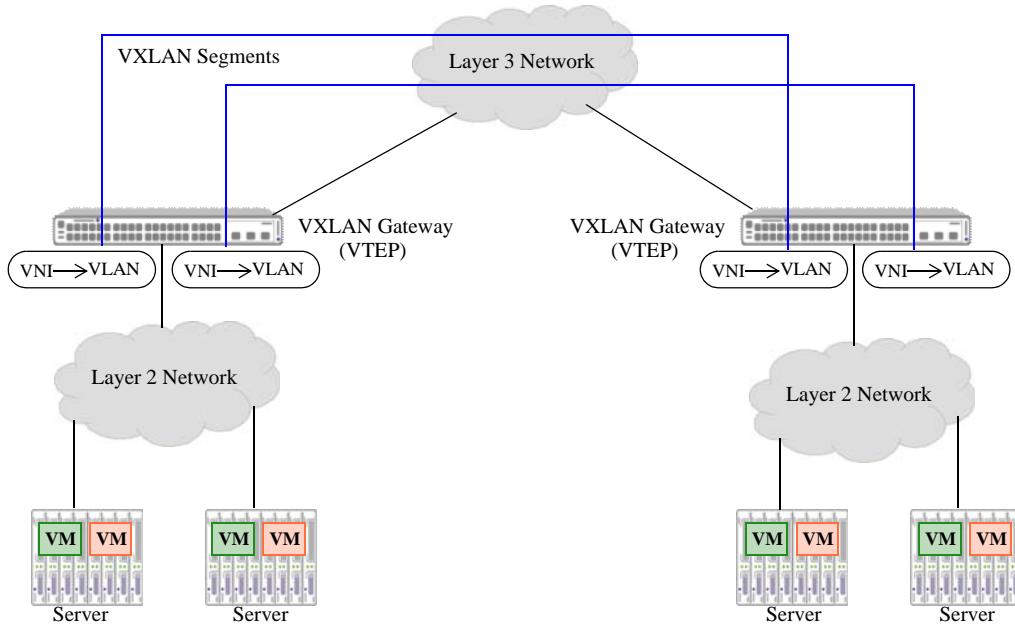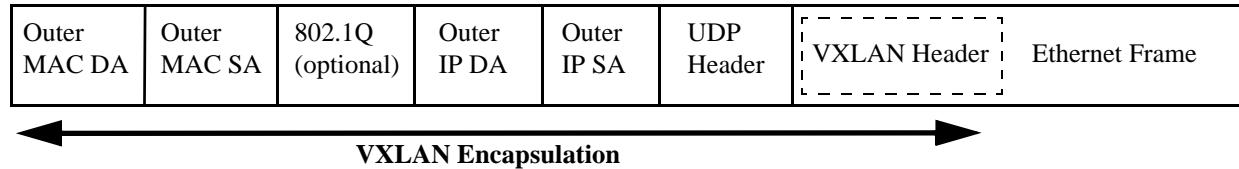The following diagram provides an example of a VXLAN deployment:

Figure 1: OmniSwitch VXLAN Gateway Example

In this example:

- The OmniSwitch OS6900-Q32 serves as a VXLAN gateway. The VTEPs on each switch initiate and terminate VXLAN tunnels to form a virtual network over the shared Layer 3 infrastructure.

- VM traffic initiated from one of the servers traverses a Layer 2 network within a specific VLAN domain and reaches the VXLAN gateway switch.

- If the gateway switch determines that the VM traffic is destined for another server or VM that is on a remote Layer 2 network, the gateway encapsulates the traffic and forwards it on the VXLAN tunnel.

- When the gateway switch receives an encapsulated VXLAN packet, the encapsulation header is removed and the original VM traffic is forwarded onto the VLAN domain for which the VM traffic was destined.

- The VXLAN gateway encapsulation and forwarding process is transparent to the VMs in this example.

- If the gateway switch determines that the VM traffic is destined for another device within the local Layer 2 network, then the traffic is bridged on the local VLAN domain. In this case, the VXLAN encapsulation and tunneling process is not used.

VXLAN Encapsulation

When an OmniSwitch VXLAN gateway receives an Ethernet MAC frame originating from a Virtual Machine (VM) that is destined for a remote VM, the switch encapsulates the frame into an IP packet for transport through a VXLAN service tunnel. The following is an example of the IP packet format after VXLAN encapsulation:

Outer MAC DA	Outer MAC SA	802.1Q (optional)	Outer IP DA	Outer IP SA	UDP Header	VXLAN Header	Ethernet Frame

VXLAN Encapsulation

The VXLAN encapsulated packet consists of the following components:

- **VXLAN Header**—Contains a VXLAN Network ID that is added to the original Ethernet frame. This ID is specified and associated with a VXLAN service ID when a VXLAN service is created. The service is then bound to a VXLAN Service Distribution Point (SDP), which tunnels the encapsulated packets.

- **UDP Header**—Contains a source port derived from the original Ethernet frame and the destination port is set to the UDP port designated for VXLAN traffic. UDP port 4789 is used by default, but this value is also configurable through the OmniSwitch CLI.

- **IP Header**—The source IP address in this header is the Loopback0 interface address for the local VXLAN Tunnel End Point (VTEP). The destination address is set to the Loopback0 interface address of the far-end VTEP for unicast communication and a multicast IP address for broadcast, unknown unicast, and multicast communication. The Loopback0 IP address and/or multicast IP address for a VTEP is specified when a VXLAN SDP is created.

- **Outer Ethernet Header**—The outer DA MAC address is the MAC address of the next hop router and the outer SA address is the MAC address of the local router. An 802.1Q value is present if the packet is tagged.

Note. VXLAN encapsulation adds 50 bytes of overhead to the original frame. VTEPs must not fragment the VXLAN packets. However, intermediate routers may fragment the encapsulated VXLAN packets due to the larger frame size, and the destination VTEP may silently ignore the fragmented packets.

VXLAN MAC Learning and Packet Forwarding

VXLAN uses source learning and flooding to build the underlying data plane required for transporting encapsulated VXLAN packets. Source learning discovers the association between a VM MAC address and the VXLAN Tunnel End Point (VTEP) IP address. IP multicast is used to flood broadcast, unknown unicast, and multicast (BUM) traffic. Specifically, Bidirectional PIM (BIDIR-PIM) is recommended because a VTEP can serve as both a source and a destination for BUM traffic.

VM-to-VM Communication with Unknown Destination

When a VM attempts to communicate with another VM (destination VM) on the same subnet, the VM sends out a Layer 2 ARP broadcast frame. In the standard VLAN environment, the switch would broadcast the frame across all switches that are reachable within the VLAN domain. However, in a VXLAN environment a Layer 2 ARP frame is encapsulated with the following:

- A VXLAN header containing a VXLAN Network Identifier (VNI). The VNI associates the ARP frame with a VXLAN segment.

- An IP header containing a source and destination IP. The source IP is the IP address of the local VTEP. The destination IP is a multicast IP address designated for BUM traffic.

- A UDP header containing a source port derived from the original ARP frame and the destination UDP port number that is designated for VXLAN traffic.

The encapsulated ARP frame is then sent out to the IP multicast group on which the VXLAN segment exists. This requires a mapping between the VXLAN VNI and the multicast group IP address. On the OmniSwitch, this mapping is achieved through the manual configuration of a VXLAN Service Distribution Point (SDP) to define a VXLAN tunnel and associate the tunnel with a multicast group IP address. A VXLAN service associated with a VNI is then manually bound to the SDP.

The destination VM sends a standard ARP response which is also encapsulated and sent through a VXLAN tunnel. However, the response is sent using IP unicast to the VTEP of the originating VM. This is possible because the destination MAC address for the ARP response is mapped to the VTEP IP address that was previously learned when the ARP request was generated. The dynamic learning of a remote MAC address association with a remote VTEP IP address is retained for future communication with traffic destined for that specific MAC address.

VM-to-VM Communication with Known Destination

When a source VTEP is aware of a remote VTEP IP (through user configuration or dynamically learned), unicast VM-to-VM communication is possible. For example, when a VM attempts to communicate with a remote VM, the local VTEP looks up the VNI to which the VM is associated. If the destination MAC address is on the same VNI segment and the destination MAC address is mapped to the remote VTEP IP, the Layer 2 frame is encapsulated with the following:

- A VXLAN header containing a VXLAN Network Identifier (VNI). The VNI associates the ARP frame with a VXLAN segment.

- An IP header containing a source and destination IP. The source IP is the IP address of the local VTEP. The destination IP is the known IP address of the remote VTEP.

- A UDP header containing a source port derived from the original frame and the destination UDP port number that is designated for VXLAN traffic.

- An outer Ethernet header containing the MAC address of the next hop router and the MAC address of the local router.

The encapsulated VXLAN packet is then forwarded towards the remote VTEP, which removes the encapsulation headers and forwards the original Layer 2 frame to the destination VM. During this process, the remote VTEP learns the mapping of the inner source MAC address to the outer source VTEP IP address. This information is retained for future unicast communication between the source and destination VM.

Multicast Forwarding

An OmniSwitch VXLAN gateway supports the transport of learned unicast and broadcast, unknown unicast, and multicast (BUM) traffic across VXLAN segments. The learned unicast traffic is directed to the destination virtual port on which the destination MAC was learned. If the traffic is sent out on a local virtual port, no encapsulation is required. If the traffic is sent out to a remote destination through a VXLAN tunnel, the traffic is encapsulated with the outer destination IP address set to the IP address of the remote VTEP. On the OmniSwitch, this is the IP address configured for the Loopback0 interface.

BUM traffic is flooded out to remote VTEPs using one of two methods: tandem or head-end replication.

- **Head-end replication**. Multicast traffic is replicated once for each receiver, encapsulated and then sent as a unicast packet to each destination. This method requires known Loopback0 IP addresses for all remote VTEPs participating in the same VXLAN segment. This is achieved through the manual configuration of a VXLAN Service Distribution Point (SDP) with a far-end unicast IP address. Head-end replication is more suited for networks where there is a low demand for multicast traffic.

- **Tandem**. This is standard IP multicast, where a multicast routing protocol (BIDIR-PIM on the OmniSwitch) forms a forwarding topology for each multicast group IP address. Each VTEP only has to signal through the multicast routing protocol their interest in the multicast groups assigned to the VNIs in which the VTEP participates. The tandem method requires manual configuration of an SDP with a multicast group IP address instead of a far-end unicast IP address.

Unicast and Multicast Routing

This implementation of VXLAN makes use of the underlying OmniSwitch routing and multicast framework. Specifically, Bidirectional PIM (BIDIR-PIM) is used to set up the multicast domain, and OSPF, RIP, IS-IS, or static routing is used to set up the routing domain for unicast paths.

VXLAN overlay networks are built on top of IP transport networks through a combination of unicast and multicast IP tunnels. IP unicast routing is a mature, widely used technology with a proven track record of success. The OmniSwitch includes support for multiple IP unicast routing protocols and components that are used to build a highly resilient IP network.

IP multicast routing is also a mature technology that provides the foundation for efficient one-to-many and many-to-many communication. The OmniSwitch includes support for several IP multicast routing protocols, in addition to IP multicast snooping/switching (IPMS). Configuring IPMS does not require an IP configuration, but the use of multicast routing does require unicast routing to prevent routing loops and the duplication of traffic. When multicast routing is configured, IPMS functionality is automatically activated on the VLANs associated with the routed IP interfaces.

As previously mentioned, BIDIR-PIM is the recommended multicast routing protocol for supporting the OmniSwitch VXLAN implementation. BIDIR-PIM is an optimization of PIM Sparse Mode (PIM-SM) that reduces the amount of state a router must maintain. This is done by eliminating source-specific (S,G) forwarding in favor of shared tree (*,G) forwarding.

Using BIDIR-PIM requires the configuration of one or more IP interfaces and Rendezvous Point (RP) information for the multicast groups in the network. PIM group mappings are either configured statically or dynamically learned through Candidate-RP advertisements using the BootStrap Router (BSR) mechanism.

VXLAN Service Components

The OmniSwitch implementation of VXLAN is achieved through the Service Manager framework. LANs and local devices are associated with a VXLAN Network ID through the configuration of service objects. This approach requires the configuration of the following service components to define the VXLAN service domain:

- **VXLAN service:** The OmniSwitch VXLAN gateway identifies each VXLAN segment that the gateway participates in with a VXLAN service ID. A VXLAN service is associated with a VXLAN Network Identifier (VNI) at the time the service is created. This type of service represents a single virtual bridge on the gateway switch.

- **Access Port**. A port or link aggregate configured as a service access port. This type of port is configured on the VXLAN gateway switch to forward traffic to or from the VXLAN overlay network.

The access port is also associated with a Layer 2 profile that specifies how to process protocol control frames received on the port.

- **Service Access Point (SAP)**—A SAP is a logical service entity (also referred to as a virtual port) that is configured on a gateway switch to bind an access port and traffic received on that port to a VXLAN service ID (and the associated VNI).

- **Service Distribution Point (SDP)**—An SDP provides a logical point at which device traffic is directed from one VXLAN gateway to another VXLAN gateway. SDPs are used to set up distributed services, which consist of at least one SAP on a local node, one SAP on a remote node, and an SDP binding the service on both nodes. An SDP serves as the VXLAN Tunnel Interface (VTI) on a gateway switch.

- **SDP Bind**—An SDP binding represents the binding of a VXLAN service instance to an SDP. The SDP then distributes the service connectivity to other VXLAN gateways.

The following diagram shows how the above components are used to construct a VXLAN overlay network and tunnel traffic through that network:
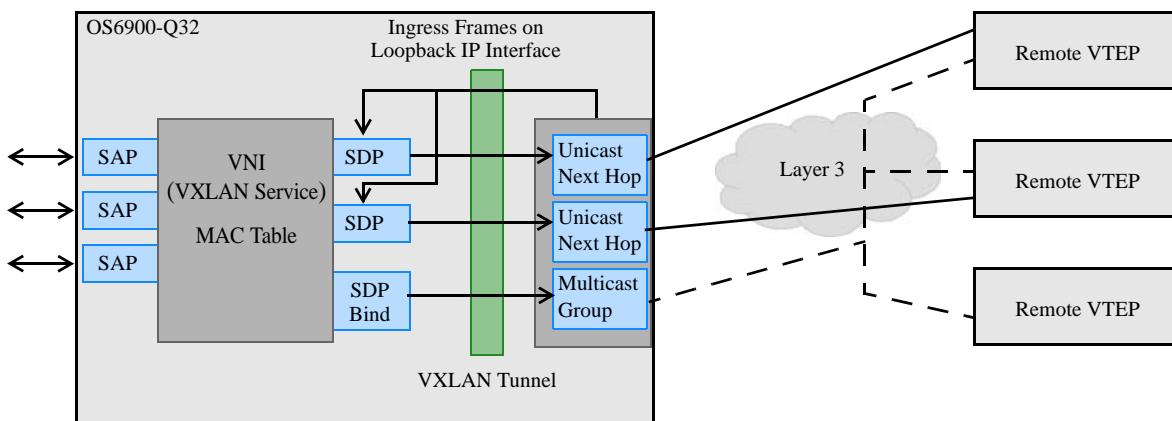
Figure 2: OmniSwitch VXLAN Service Components

In this example diagram:

- The OS6900-Q32 is operating as a VXLAN gateway, which is also referred to as a VXLAN Tunnel End Point (VTEP). The source IP address for the VTEP is the IP address configured for the Loopback0 interface on the switch.

- SAPs are configured on service access ports and define which VLAN IDs will enter the service domain. A VXLAN service associated with a VNI is also bound to the SAP. This is the service that will carry the SAP traffic through the VXLAN domain.

- On the same gateway switch, VXLAN SDPs are defined and bound to the VXLAN services. The SDPs operate as VXLAN tunnel interfaces (VTIs) on which device traffic is encapsulated and sent through the VXLAN tunnel. Device traffic received on the VTI is stripped of the encapsulation headers and then forwarded onto the destination VLAN domain.

- The remote VTEPs shown in this diagram can be other OmniSwitch 6900-Q32 gateway switches or a virtual switch on a server.

Interaction With Other Features

This section contains important information about Virtual eXtensible Local Area Network (VXLAN) 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.

Hardware

The VXLAN functionality is supported only on an OmniSwitch 6900-Q32. This means that VXLAN services, SAPs, and SDPs are only configurable on an OmniSwitch 6900-Q32 switch. In addition, VXLAN is not supported on ports or link aggregates that have the following features enabled:

Features	
Individual linkagg member port	UNP is a property of a link aggregate logical port and not a property of an individual member port.
Tagged port	Tagged VLAN-port associations (VPAs) are created by UNP after learning a MAC address.
VLAN stacking UNI port	VXLAN is one type of service. Other types of services are not configurable on the same port.
VLAN stacking NNI port	VXLAN is an access port function; not supported on the network side.
ERP port:	ERP can control the forwarding behavior of the port (similar to STP)
Port mirroring port	Carries the traffic from other ports; does not carry any original traffic.
Ports configured with a static MAC address	If static MAC addresses are configured, then VXLAN does not have full control of the traffic on that port.
Ports with MAC address learning disabled	VXLAN needs to learn the MAC address to determine how to process the Layer 2 frames and encapsulated VXLAN packets. If learning is disabled, this functionality is not achieved.
Service Manager Access Port	If a port is already a Service Manager access port, then enabling UNP does not flush the services already learned on the port. It may cause an inconsistent state across modules.
FCoE Initiation Protocol Snooping (FIPS)	FIPS program policies are based on the notion of a physical port, whereas VXLAN is applied through virtual ports. Both cannot coexist on the same port.

Link Aggregation

- Both static and dynamic link aggregates are configurable as service access ports.

- Note that a link aggregate must consist of all access ports or all network ports. VXLAN functionality is not supported on link aggregates that consist of a mixture of access ports and standard switch ports.

Loopback0 IP Interface

Configuring the Loopback0 IP interface is required to allow the OmniSwitch to serve as a VXLAN Tunnel End Point (VTEP), also referred to as a VXLAN gateway. The IP network address for this interface is used as the source IP address for the VTEP, which is used in VXLAN encapsulation.

- A routing protocol or static route configuration is required to ensure reachability between all the VTEP IPs in the VXLAN overlay network.

- The Loopback0 IP address can be redistributed using a route-map to propagate reachability in the Layer 3 cloud.

- A multicast routing configuration is also required to ensure that all the VTEPs join the associated multicast group.

Bidirectional Protocol Independent Multicast (BIDIR-PIM)

- When a VXLAN service is configured to use the Tandem multicast mode, a multicast infrastructure is required in the network. In an OmniSwitch network, using BIDIR-PIM to provide this infrastructure is recommended.

- Each OmniSwitch that will serve as a VXLAN gateway must be configured as a VXLAN Tunnel End Point (VTEP). A VTEP will act as a multicast speaker and a multicast receiver at the same time.

- Any VXLAN service that is configured to use the Head-end multicast mode does not require the underlying multicast infrastructure, so BIDIR-PIM is not required in this case.

- BIDIR-PIM is the only multicast protocol supported with VXLAN.

Quality of Service (QoS)

- The priority assignment of a user frame is determined at an access point. A Service Access Point (SAP) on an access port can be configured to as trusted or untrusted. If a SAP is configured to as trusted, then the internal priority for ingress traffic on that SAP is derived from the tagged or NULL tagged ingress packet priority or from the default port priority if the ingress packet is untagged. If a SAP is untrusted, then the internal priority can be user configured.

- QoS performs the following actions on ports configured as access ports:
  - Access ports are automatically trusted and the default classification is set to 802.1p.
  - The trust status and classification are not user-configurable on access ports.
  - All QoS CLI configuration is blocked on access ports.This includes physical ports and ports that are members of a link aggregate.
  - Untagged L2 control packets (such as BPDU, GVRP, AMAP) are always tunneled (if enabled) through the VXLAN domain with the default EXP bits set at 7, so that they can arrive at the destination at the highest priority of 7. Trusted and untrusted SAPs configured on access ports will not affect the priority assignment for Layer 2 control packets.

- QoS priority (802.1p) is applied as follows to trusted and untrusted SAPs:

SAP Configuration	Allowed Configuration	
Tagged (VLAN 1–4094)	Trusted	Tagged traffic priority derived from tags.
	Untrusted	Tagged traffic priority configured by user.
QinQ (outer VLAN 1–4094)	Trusted	Tagged traffic priority derived from outer tags.
	Untrusted	Tagged traffic priority configured by user.
Wild Card	Trusted	Tagged traffic priority derived from tags. Untagged traffic Port default (PRI 0).
	Untrusted	Tagged/ traffic priority configured by user
Untagged	Trusted	Untagged Traffic Port default (PRI 0)
	Untrusted	Priority configured by user.

– By default, a SAP is trusted with best effort priority (0)

– A SAP can be dynamically changed to trusted or untrusted without an admin down of the SAP.

– A SAP priority may only be set when a SAP is untrusted.

– When a SAP is changed from untrusted to trusted, any previously assigned priority is reset with best effort priority (0).

– A trusted SAP that defines a double-tagged encapsulation (QinQ) will use the outer VLAN tag to determine the priority of the frame.

Universal Network Profiles (UNP)

Integration with Virtual Machine Network Profiles (vNPs) to support device discovery and mobility. The UNP feature supports two types of profiles: VLAN and service. A service profile can be configured to classify traffic for VXLAN or SPB tunneling.

The OmniSwitch supports both a VLAN and service domain for traffic classification.

● The VLAN domain is identified by a VLAN ID. In the VLAN domain, each VLAN is accessed through a physical port. Each physical port can have more than one VLAN attached. UNP VLAN classification associates a MAC address to a specific VLAN on a physical UNP bridge port.

● The service domain is identified by one of the following:

– A VXLAN Network Identifier (VNI), which is associated with a Service Manager service ID to represent a virtual forwarding instance (VFI).

– A Shortest Path Bridging (SPB) service instance identifier (I-SID), which is associated with a Service Manger service ID to represent a VFI.

In the service domain, each VFI is accessed through a virtual port, referred to as a Service Access Point (SAP). UNP service classification associates a MAC address to a SAP.

Dynamic Service Access Points

A UNP service profile can trigger the dynamic creation of a SAP when traffic received on a UNP access port is classified and assigned to that profile. If the service (VXLAN or SPB) that the SAP is associated with does not exist, the service is also dynamically created.

Allowing incoming traffic to trigger dynamic SAP creation reduces the amount of manual configuration required. In addition, no other protocols are required on the switch or host device to support this functionality.

UniDirectional Link Detection (UDLD)

UDLD protocol control frames (destination MAC address is 01:00:0c:cc:cc:cc) are processed as follows:

UDLD Status	User Access Port	Network Port (Tagged)	Network Port (Untagged)	Legacy
Globally disabled	tunnel	tunnel	discard	tunnel
Globally enabled	tunnel	tunnel	discard	drop
Enabled on port	peer	tunnel	peer	peer

Source Learning

- Source learning is used to discover the association between a VM source MAC address and a VTEP IP address.

- The "vxlan" domain parameter is available to filter the MAC address table output for VXLAN traffic.

Virtual Chassis (VC)

The VXLAN functionality on an OS6900-Q32 or an OS6900-X72 is supported in a mixed VC setup (the OS6900-Q32 or OmniSwitch 6900-X72 with other OS6900 models). Consider the following guidelines when configuring VXLAN in a mixed VC setup:

- Configure service access ports on any of the OS6900 models.

- Configure network ports only on the OS6900-Q32 or OS6900-X72. This is done to make sure that all the routing and multicast functionality required to support the VXLAN tunnel service is only provided through the OS600-Q32 or OS6900-X72 chassis.

- ECMP routes are not supported for VXLAN tunnels when the service access ports are on OS6900 models other than the OS6900-Q32 or OS6900-X72. Configure all routing for VXLAN tunnels to avoid ECMP on the network ports. Otherwise, user traffic that originates on the other OS6900 models cannot be forwarded out the VXLAN tunnel ports on the OS6900-Q32 or OS6900-X72.

- Configure SAPs on any of the OS6900 models.

- Table sizes will synchronize to the least common denominator.

- Setting a higher chassis priority for the OS6900-Q32 or OS6900-X72 is recommended to give the OS6900-Q32 or OS6900-X72 a higher precedence for the VC master selection.

Configuring a mixed VC setup incorrectly could lead to unexpected behavior. Configuration restrictions are not imposed through the implementation of this feature. It is strictly up to the administrator to configure VXLAN in a mixed VC setup according to the recommended guidelines.

The following table contains the maximum values for resources in a mixed VC setup:

Resource	OS6900-X/T and OS6900-Q32/X72
VXLAN service instances	1K
Virtual port (SAPs and network binds)	4K/8K
Service instance counters	1K
Virtual port counters	4K/8K
Tunnel Terminations (MPLS_ENTRY)	8K
Next hops	8K

If a VC setup consists of all OS6900-Q32 switches or all OS6900-X72 switches, the configuration guidelines and maximum resource values described in this section do not apply.

VXLAN Snooping

A software mechanism used to identify and inspect VXLAN packets (encapsulated VM frames) received on OmniSwitch interfaces configured as VM snooping ports. VM snooping builds a database of the VM information obtained during the snooping process. This information is then available for administrators to use for traffic managing, monitoring, and troubleshooting.

VRF

The VXLAN functionality is supported only in the default VRF instance.

Configuring a VXLAN Gateway

Configuring a VXLAN gateway (also referred to as a VXLAN Tunnel End Point) requires several steps. These steps are outlined here and further described throughout this section. For a brief tutorial on configuring VXLAN, see "Quick Steps for Configuring a VXLAN Gateway" on page 4-4.

1 Configure the required routing and multicast framework. Although any IGP routing protocol can be used to build the routing domain for unicast paths, static routes are recommended for performance and stability. Configuring BIDIR-PIM is used to build the multicast domain.

2 Create a VXLAN service. A Service Manager service ID is associated with a VXLAN Network ID and a service access point (SAP) to identify the device traffic that the service will tunnel through the VXLAN segment. See "Creating a VXLAN Service" on page 4-17.

3 Configure service access ports. One or more access ports are associated with a Service Access Point to identify to the service which ports will receive device traffic that the service will process for tunneling through the VXLAN segment. When an access port is associated with a SAP, the SAP parameter attributes are applied to traffic received on the access port. See "Configuring Service Access Ports" on page 4-21.

4 Optionally define access port profile attributes. A default Layer 2 profile is automatically assigned to an access port at the time the port is configured as an access port. This profile determines how control frames received on the port are processed. It is only necessary to configure a Layer 2 profile if the default attribute values are not sufficient. See "Configuring Layer 2 Profiles for Access Ports" on page 4-21.

5 Configure a VXLAN Service Access Point (SAP). A SAP binds a VXLAN service to an access port and defines which device traffic is tunneled through the service. Each SAP is associated to one service ID, but a single service can have multiple SAPs to which it is associated. See "Configuring Service Access Points (SAPs)" on page 4-19.

6 Configure a VXLAN Service Distribution Point (SDP). An SDP serves as a VXLAN tunnel interface on a VXLAN gateway switch and defines a unicast or multicast path for a VXLAN segment. See "Configuring Service Distribution Point (SDPs)" on page 4-27.

7 Bind a VXLAN service to an SDP. An SDP defines a path, and binding a VXLAN service to an SDP defines the path on which service traffic is forwarded. SDPs and SDP bindings require manual configuration. See "Binding VXLAN Services to SDPs" on page 4-28.

8 Optionally define the UDP destination port for VXLAN packets. By default, the well-known UDP port number 4789 is used in VXLAN packets. To change this number, see "Configuring the UDP Port for a VXLAN Gateway" on page 4-30.

Refer to the "VXLAN Gateway Configuration Examples" on page 4-31 to see how the these configuration steps are used to build a VXLAN overlay network.

Creating a VXLAN Service

A VXLAN service is identified by a service ID number, which is bound to a VXLAN Network Identifier (VNI). The **service vxlan** command is used to create a VXLAN service. For example, the following command creates VXLAN service 10 and binds the service to VNI 2300:

```
-> service 10 vxlan vnid 2300
```

The VNI number specified with this command creates a new segment ID that is bound to the specified service ID. The service will then carry all VXLAN traffic associated with the specified segment ID.

Modifying Default VXLAN Service Parameters

The following VXLAN service parameter values are set by default at the time the service is created. If necessary, use the specified commands to change the default values.

Parameter Description	Command	Default
Service description.	**service description**	None
Administrative status for statistics collection.	**service stats**	Disabled
Multicast mode	**service multicast-mode**	hybrid
VLAN translation	**service vlan-xlation**	Disabled
Administrative status of the service	**service admin-state**	Enabled

Refer to the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the above parameters and related commands.

Using VLAN Translation

VLAN translation refers to the egress translation of VLAN tags on service access points (SAPs). When enabled for a service, the VLAN tags for outgoing VM frames on SAPs associated with that service are processed according to the local SAP configuration (the SAP on which the frames will egress) and not according to the configuration of the SAP on which the frames were received.

- If the local SAP is configured for untagged traffic (*slot/port*:0), the egress traffic is always sent out as untagged.

- If the local SAP is configured for 802.1q-tagged traffic (*slot/port:ctag*), the egress traffic is single-tagged with the tag value specified by the *ctag* (customer VLAN tag) value.

- If the local SAP is configured for double-tagged traffic (*slot/port:outer_tag.: inner_tag*), the egress traffic is double-tagged with the tag values specified by the *outer_tag* and *inner_tag* values.

When VLAN translation is disabled, frames simply egress without any modification of the VLAN tags. In other words, the frames are transparently bridged without tag modification.

The following table shows the required translation (tag is added or replaced) that takes place when the egress SAP configuration is applied to the possible frame types (untagged, tagged, double-tagged). Note that in this table the terms "ITAG" and "OTAG" refer to inner tag and outer tag, respectively.

	Egress SAP (action required based on SAP type)		
	Untagged SAP	**Single Tagged SAP**	**Double-Tagged SAP**
Incoming Frame	Remove OTAG Remove ITAG	Replace OTAG **Note**: Replace = implicit add Remove ITAG	Replace OTAG **Note**: Replace = implicit add Add/Replace ITAG
Untagged	No tags, no action taken	Add the SAP OTAG	Add the SAP OTAG Add the SAP ITAG.
Single-tagged	Remove the OTAG	Replace the OTAG	Add ITAG Replace OTAG
Double-tagged	Remove the ITAG Remove the ITAG	Remove the ITAG Replace the OTAG	Replace ITAG Replace OTAG

Enabling VLAN translation is required at two different levels: first at the access port level and then at the service level. This activates VXLAN translation for all of the SAPs on an access port that belong to the same service.

To enable translation at the service level, use the **service vlan-xlation** command. For example:

```
-> service 10 vlan-xlation enable
```

To enable VLAN translation for all services, use the **all** parameter with the same command. For example:

```
-> service all vlan-xlation enable
```

To disable VLAN translation, use the **service vlan-xlation** command with the disable parameter. For example:

```
-> service 10 vlan-xlation disable
-> service all vlan-xlation disable
```

To enable VLAN translation at the port level, use the **service access vlan-xlation** command. For example:

```
-> service access port 1/11 vlan-xlation enable
```

See "Configuring Service Access Ports" on page 4-21 for more information.

Enable the Service

By default, the VXLAN service is disabled when the service is created. Once the service is created and any optional service parameters are configured, use the **service admin-state** command with the **enable** option to enable the service. For example:

```
-> service 10 admin-state enable
```

To disable the service, enter the following command:

```
-> service 10 admin-state disable
```

Deleting a VXLAN Service

Before deleting a service from the switch configuration, disable the administrative status of the service. Once this is done, use the **no** form of the **service vxlan** command to delete the service. For example:

```
-> no service 10
```

Verifying the VXLAN Service Configuration

To view the VXLAN service configuration for the switch, use the **show service** command with the **vxlan** parameter option. For example:

```
-> show service vxlan
VxLAN Service Info
                                SAP   Bind                    MCast
ServiceId  Adm  Oper Stats Count Count Vnid               Mode
-----------+----+----+-----+-----+-----+--------------+--------------------------
2          Up   Up    Y    2     1     2000 (0.7.208)     -     Headend
3          Up   Up    Y    2     1     3000 (0.11.184)    -     Headend
10         Up   Up    Y    1     3     1000 (0.1.56)   224.2.1.1 Tandem
32769*     Up   Up    Y    1     3     5000 (0.19.136) 225.1.1.1 Tandem

Total Services: 4
```

To view the configuration for an individual service, use the **show service** command and specify the VXLAN service ID. For example:

```
-> show service 10
VxLAN Service Detailed Info
  Service Id      : 10,               Description      : VXLAN Svc VNID 1000,
  VNID            : 1000 (0.1.56),
  Multicast-Mode  : Tandem,
  Admin Status    : Up,               Oper Status      : Up,
  Stats Status    : Yes,              Vlan Translation : No,
  Service Type    : VxLAN,            Allocation Type  : Static,
  MTU             : 9194,             Def Mesh VC Id   : 1,
  SAP Count       : 1,                SDP Bind Count   : 3,
  Ingress Pkts    : 0,                Ingress Bytes    : 0,
  Egress Pkts     : 0,                Egress Bytes     : 0,
  Mgmt Change     : 10/19/2014 13:20:42,  Status Change    : 10/19/2014 13:20:42
```

Configuring Service Access Points (SAPs)

A SAP identifies the location where traffic enters a VXLAN segment, the type of traffic to service, parameters to apply to the traffic, and the service that will process the traffic for tunneling through the VXLAN segment.

Configuring a SAP requires several steps. These steps are outlined here and further described throughout this section:

- Configure switch ports or link aggregates as service access ports.

- Configure Layer 2 profiles to determine how control packets are processed on access ports.

- Create a SAP by associating a SAP ID with a service ID. A SAP ID is comprised of an access port and an encapsulation value, which is used to identify the type of traffic (untagged, single-tagged, or double-tagged) to map to the associated service.

SAP Configuration Guidelines

Consider the following when configuring a SAP:

- A SAP is a unique local entity for any given device. The same SAP ID value can be used on other VXLAN gateway switches.

- There are no SAPs configured by default; explicit configuration of a SAP is required.

- A SAP is administratively disabled at the time the SAP is created.

- When a SAP is deleted, all configuration parameters for the SAP are also deleted.

- A SAP is owned by and associated with the service that was specified at the time the SAP was created.

- Multiple SAPs with different service types, such as a VXLAN or a Shortest Path Bridging (SPB) service, are allowed on the same service access port. For example, the following **show service access** command output shows two SAPs for port 2/1/30: one SAP bound to a VXLAN service and the other SAP bound to an SPB service:

```
-> show service acc port 2/1/30 sap
Legend: * denotes a dynamic object
                                                           Vlan
Identifier     Adm  Oper Stats T:P  ServiceId   Isid/Vnid Xlation Sap Description
-------------+----+----+-----+----+-----------+---------+-------+---------------
sap:2/1/30A:0        Down Down  N    Y:x  20       1500      N       -
sap:2/1/30A:5        Up   Down  N    Y:x  10       23000     N       -

Total SAPs: 2
```

- If a port is administratively shutdown, all SAPs on that port become operationally out of service.

- Both fixed ports and link aggregates are configurable as access ports. Only access ports are associated with SAPs.

- Bridging functionality is not supported on access ports or link aggregates.

- Configuring multiple SAPs on an access port that map different VLAN tags to the same service can cause a MAC move when the same MAC address ingresses the access port with different VLAN tags. For example, a MAC has two flows tagged with VLAN 10 and VLAN 20 that ingress access port 1/1 and both flows are mapped to service 100.

```
-> service 100 sap port 1/1:10
-> service 100 sap port 1/1:20
```

To avoid the MAC move in this scenario, use one of the following alternative SAP configurations.

Configure the SAPs with different services:

```
-> service 100 sap port 1/1:10
-> service 200 sap port 1/1:20
```

Configure a default SAP to classify both flows into the same service:

```
-> service 100 sap port 1/1:all
```

See "Creating the Service Access Point" on page 4-23 for more information.

Configuring Service Access Ports

Each SAP is comprised of a service access port or link aggregate and an encapsulation type value. Access ports are configured on a VXLAN gateway switch to forward traffic to or from a VXLAN segment. Traffic received on these ports is classified for one or more SAPs and forwarded onto the intended destination by the associated VXLAN service.

To configure a port or link aggregate as an access port, use the **service access** command. For example, the following command configures port 1/2 and link aggregate 100 as access ports:

```
-> service access port 1/2
-> service access linkagg 100 description "Server Access Port"
```

In the link aggregate example, an optional **description** parameter is used to add information to the access port. This parameter is available at the time the port is configured as an access port or to change or remove a description from an existing access port. For example, the following command removes the description from link aggregate 100:

```
-> service access linkagg 100 no description "Server Access Port"
```

After the description is removed, link aggregate 100 continues to operate as a service access port. To revert an access port back to a regular switch port or link aggregate, use the **no** form of the **service access** command. For example:

```
-> no service access port 1/2
-> no service access linkagg 100
```

VLAN Translation on Access Ports

VLAN translation refers to the egress translation of VLAN tags on service access points (SAPs). For more information about how VLAN translation is applied, see .

By default, VLAN translation is disabled on access ports. Enabling VLAN translation on an access port implicitly enables translation for all SAPs associated with that port. However, translation must also be enabled for the services associated with these SAPs. This ensures that all SAPs associated with a service will apply VLAN translation.

To enable VLAN translation on an access port, use the **service access vlan-xlation** command with the **enable** option. For example:

```
-> service access port 1/3 vlan-xlation enable
-> service access linkagg 10 vlan-xlation enable
```

To disable VLAN translation on an access port, use the **service access vlan-xlation** command with the **disable** option. For example:

```
-> service access port 1/3 vlan-xlation disable
-> service access linkagg 10 vlan-xlation disable
```

Configuring Layer 2 Profiles for Access Ports

A Layer 2 profile determines how ingress control frames on an access port are processed. When a port is configured as an access port, a default Layer 2 profile (**def-access-profile**) is applied to the port with the following default values for processing control frames:

Protocol	Default
STP	tunnel
802.1x	drop
802.1ab	drop
802.3ad	peer
GVRP	tunnel
MVRP	tunnel
AMAP	discard

If the default profile values are not sufficient, use the **service l2profile** command with the **tunnel**, **discard**, and **peer** options to create a new profile. For example, the following command creates a profile named "DropL2":

```
-> service l2profile DropL2 stp discard gvrp discard 802.1ab discard
```

Consider the following when configuring Layer 2 profiles:

- Not all of the control protocols are currently supported with the **peer**, **tunnel**, and **discard** parameters. Use the following table to determine the parameter combinations that are supported:

Protocol	Reserved MAC	peer	discard	tunnel
STP	01-80-C2-00-00-00	no	yes	yes
802.1x	01-80-C2-00-00-03	no	yes	yes
802.1ab	01-80-C2-00-00-0E	yes	yes	yes
802.3ad	01-80-C2-00-00-02	yes	no	no
GVRP	01-80-C2-00-00-21	no	yes	yes
MVRP	01-80-C2-00-00-21	no	yes	yes
AMAP	00-20-DA-00-70-04	yes	yes	no

- When a profile is created, the new profile inherits the default profile settings for processing control frames. The default settings are applied with the new profile unless they are explicitly changed. For example, the profile "DropL2" was configured to discard STP, GVRP, and 802.1ab frames. No other protocol settings were changed, so the default settings still apply for the other protocols.

- Remove any profile associations with access ports before attempting to modify or delete the profile.

To delete a Layer 2 profile, use the **no** form of the **service l2profile** command. For example, the following command deletes the "DropL2" profile:

```
-> no service l2profile DropL2
```

Use the **show service l2profile** command to view a list of profiles that are already configured for the switch. This command also displays the attribute values for each profile.

Assigning Layer 2 Profiles to Access Ports

After a Layer 2 profile is created, it is then necessary to assign the profile to an access port or link aggregate. When this is done, the current profile associated with the port is replaced with the new profile.

The **service access l2profile** command is used to assign a new profile to an access port. For example, the following command assigns the "DropL2" profile to access port 1/4:

```
-> service access port 1/4 l2profile DropL2
-> service access linkagg 5 l2profile DropL2
```

To change the profile associated with the access port back to the default profile (**def-access-profile**), use the **default** option with the **service access l2profile** command. For example:

```
-> service access port 1/4 l2profile default
-> service access linkagg 5 l2profile DropL2
```

Use the **show service access** command to display profile associations for access ports.

Verifying the Access Port Configuration

To view the access port configuration for the switch, use the **show service access** command. For example:

```
-> show service access
Port       Link  SAP     SAP    Vlan
Id         Status Type   Count  Xlation L2Profile          Description
---------+------+-------+------+-------+------------------+---------------------
1/11      Up    Manual  100    N      def-access-profile
1/12      Up    Manual  100    N      def-access-profile
1/13      Down  Dynamic 100    N      def-access-profile  UNP Dynamic Access Port
1/14      Down  Manual  100    N      def-access-profile

Total Access Ports: 4
```

Creating the Service Access Point

Each VXLAN service is bound to at least one Service Access Point (SAP). A SAP identifies the point at which traffic enters the VXLAN segment. A SAP is one of the required service components that is needed to designate an OmniSwitch as a VXLAN gateway device.

A SAP is created by associating a SAP ID with a VXLAN service. A SAP ID is comprised of an access port and an encapsulation value that is used to identify the type of device traffic (untagged, single-tagged, or double-tagged) to map to the associated service.

The **service sap** command is used to configure a SAP. This command specifies the VXLAN service ID number and the SAP ID (slot/port:encapsulation). The following parameter values are used with this command to specify the encapsulation value:

SAP Encapsulation Value	Device Traffic Serviced
0 (null)	All untagged packets; tagged packets are dropped.
all	All tagged and untagged packets not already classified into another SAP*
qtag	Only traffic 802.1q-tagged with the specified VLAN ID.
outer_*qtag.innger_qtag*	Only traffic double-tagged (QinQ) with the specified outer and inner VLAN IDs.

*Note that the **:all** (wildcard) parameter is also configurable as the inner tag value for double-tagged frames (for example, "10:all" specifies double-tagged packets with an outer tag equal to 10 and an inner tag with any value).

The following **service sap** command example creates a SAP that will direct ingress traffic on access port 1/4 that is tagged with VLAN ID 50 to service 100:

```
-> service 100 sap port 1/4:50 description "Server1 VMs to VXLAN100 VLAN 50"
```

In the above example, the 1/4:50 designation is referred to as the SAP ID or the encapsulation ID. This means that if no other SAPs are configured for port 1/4, then any ingress traffic on that port is dropped if the traffic is not tagged with VLAN 50.

Configuring Multiple SAPs on the Same Access Port

It is possible to configure more than one SAP for the same access port, which provides a method for segregating incoming traffic into multiple services. For example, the following SAP configuration for port 2/3 sends incoming traffic to three different services based on the VLAN tags of the frames received:

```
-> service 2000 sap port 2/3:all
-> service 200 sap port 2/3:100
-> service 1000 sap port 2/3:100.200
```

In this example,

- Frames double-tagged with 100 (outer tag) and 200 (inner tag) are sent on service 1000.

- Frames single-tagged with VLAN 100 are sent on service 200.

- All other frames (those that are not single-tagged with 100 or double-tagged with 100 and 200) are sent on service 2000.

The following SAP ID classification precedence is applied when there are multiple SAPs for one access port:

1 Double-tagged (Outer VLAN + Inner VLAN) - Highest

2 Double-tagged (Outer VLAN + all)

3 Single-tagged (VLAN)

4 Single-tagged (wildcard)

5 Untagged - Lowest.

Configuring a Range of Ports or VLAN IDs for the SAP ID

It is possible to specify a range of service access ports or a range of VLAN IDs when configuring a SAP. For example, the following command configures a range of access ports and VLAN IDs for the SAP ID:

```
-> service 10 sap port 1/11-20:21-30 description "SAPs on ports 1/11 to 1/20
with tag 21 to 30"
```

In this example, any ingress traffic on access ports 1/11 through 1/20 that is tagged with a VLAN ID in the range of 21 through 30 is classified into the VXLAN SAP. The following command provides an example of specifying a range of link aggregate access ports:

```
-> service 20 sap linkagg 1-10:100 un-trusted description "Untrusted SAPs for
lag 1-10 with tag 100"
```

When specifying a range of VLAN IDs with this command on an OmniSwitch 6900-Q32, consider the following guidelines:

- Each service access port supports a maximum of 8 SAPs that are created with a range of VLANs.

- A total of 255 service access ports can support a range of VLANs at any given time.

- The range of VLANs specified for the SAPs of a service access port cannot overlap each other.

- For double-tagged SAPs, separate range commands can be specified for both outer and inner VLAN tags.

- The outer VLAN range space is shared with the same space as QTAG SAP. The combined limit is 8 unique ranges per service access port. This is in addition the 8 unique inner VLAN range per service access port.

Modifying the Default SAP Parameters

The following parameter values are set by default at the time the SAP is created. If necessary, use the specified commands to change the default values.

Parameter Description	Command	Default
SAP description.	**service sap description**	None
SAP trust mode	**service sap trusted**	Trusted
Administrative status for statistics collection.	**service sap stats**	Disabled
Administrative status for the SAP	**service sap admin-state**	Enabled

Refer to the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the command parameters.

Configuring the SAP Trust Mode

The **service sap trusted** command is used to configure the trust mode for a SAP. A trusted SAP can accept 802.1p values in incoming packets; an untrusted SAP will set any 802.1p values to zero in the incoming packets, unless an 802.1p value is configured with this command.

Note that untagged Layer 2 control packets (for example, BPDU, GVRP, and AMAP) are always tunneled (if enabled) through the VXLAN segment with the default EXP bits set to 7, so that they can arrive at the destination bridge at the highest COS queue of 7. As a result, trusted and untrusted SAPs configured on the access ports will not affect the Layer 2 control packets received on the access ports.

By default, a SAP is trusted with the priority set to best effort (zero). Use the **service sap trusted** command with the **un-trusted** and **priority** options to change the SAP mode to untrusted. For example:

```
-> service 100 sap 1/4:50 untrusted priority 7
```

When a SAP is trusted, the priority value contained in tagged device packets is used; untagged packets are assigned the default priority value (zero). When a SAP is untrusted, the priority value configured for the SAP is assigned to both tagged and untagged customer packets.

Enabling/Disabling the SAP

By default, a SAP is enabled at the time the SAP is created. To disable the SAP administrative status, use the **service sap admin-state** command. For example:

```
-> service 100 sap port 1/4:50 admin-state disable
-> service 200 sap linkagg 5:all admin-state disable
```

To enable the SAP, enter the following command:

```
   -> service 100 sap port 1/4:50 admin-state enable
   -> service 200 sap linkagg 5:all admin-state enable
```

Deleting the SAP

When a SAP is administratively disabled, the SAP configuration is not removed from the switch. To delete a SAP from the switch configuration, use the **no** form of the **service sap** command. For example:

```
   -> no service 100 sap port 1/4:50
   -> no service 200 sap linkagg 5:all
```

Verifying the SAP Configuration

A SAP is a type of virtual port that is associated with a VXLAN service. To determine the SAP configuration for a specific service, use the **show service ports** command to view the virtual ports associated with a specific service. For example:

```
-> show service 1 ports
Legend: (*) dyn unicast object
VxLAN Service 1 Info
  Admin : Up,        Oper  : Up,    Stats      : Y,        VlanXlation : Y,
  VNID  : 1000 (0.1.56),            MCast-Mode : Tandem,   MCast-Group : 224.2.1.1


                             Sap Trusted:Priority/
Identifier      Adm  Oper Stats Sdp FarEnd Addr       Intf     Description
----------------+----+----+-----+--------------------+--------+------------------
sap:1/3:0       Up   Up   N            Y:x            1/20      -
sap:1/3:10      Up   Up   N            Y:x            1/20      -
sdp:32770:1*    Up   Up   Y    10.10.10.2            1/1/1     PIM Group 224.2.1.1
sdp:32771:1*    Up   Up   Y    10.10.10.3            1/1/2     PIM Group 224.2.1.1
sdp:32772:1*    Up   Up   Y    10.10.10.4            1/1/1     PIM Group 224.2.1.1


Total Ports: 5
```

To view the configuration information for a specific VXLAN network identifier, use the **show service ports** command with the **vnid** parameter option. For example:

```
-> show service vnid 1000 ports
Legend: (*) dyn unicast object
VxLAN Service 1 Info
  Admin : Up,        Oper  : Up,    Stats      : Y,        VlanXlation : Y,
  VNID  : 1000 (0.1.56),            MCast-Mode : Tandem,   MCast-Group : 224.2.1.1


                             Sap Trusted:Priority/
Identifier      Adm  Oper Stats Sdp FarEnd/Group Addr Intf     Description
----------------+----+----+-----+--------------------+--------+------------------
sap:1/3:0       Up   Up   N            Y:x            1/20      -
sap:1/3:10      Up   Up   N            Y:x            1/20      -
sdp:10:1        Up   Up   Y    224.2.1.1             -         PIM Group 224.2.1.1
sdp:32770:1*    Up   Up   Y    10.10.10.2            1/1/1     PIM Group 224.2.1.1
sdp:32771:1*    Up   Up   Y    10.10.10.3            1/1/2     PIM Group 224.2.1.1
sdp:32772:1*    Up   Up   Y    10.10.10.4            1/1/1     PIM Group 224.2.1.1


Total Ports: 5
```

Configuring Service Distribution Point (SDPs)

An SDP serves as a VXLAN Tunnel Interface (VTI) on a switch. Configuring an SDP component is required to designate an OmniSwitch as a VXLAN gateway device and also identifies the switch as a VXLAN Tunnel Endpoint (VTEP).

A VXLAN SDP is manually configured and provides a unicast or multicast path between VTEPs. To configure an SDP on the switch, use the **service sdp vxlan** command. For example, the following command creates a VXLAN SDP that will forward unicast traffic:

```
-> service sdp 20 vxlan far-end 10.10.10.2 description "Unicast to NodeB"
```

In this example, the **far-end** option is used to specify a single destination unicast IP address that identifies a far-end VTEP node to which the SDP will tunnel traffic. This IP address is the Loopback0 interface address configured on every switch that serves as a VTEP.

In the following example, a multicast SDP is configured by using the **multicast-group** parameter option:

```
-> service sdp 10 vxlan multicast-group 224.2.1.1 ttl 20 description "PIM Group
224.2.1.1"
```

This example command specifies the multicast group address to which all broadcast, unknown unicast, and multicast (BUM) traffic is sent. All VTEP nodes that subscribe to the same multicast group through this type of SDP will receive the same traffic. Note that the **ttl** parameter value specified with this command is included in the IP header of the VXLAN encapsulation header.

Enabling/Disabling an SDP

By default, an SDP is enabled at the time the SDP is created. To disable the SDP administrative status, use the **service sdp vxlan** command with the **admin-state** parameter option. For example:

```
-> service sdp 10 admin-state disable
-> service sdp 20 admin-state disable
```

To enable an SDP, enter the following commands:

```
-> service sdp 10 admin-state enable
-> service sdp 20 admin-state enable
```

Deleting an SDP

When an SDP is administratively disabled, the SDP configuration is not removed from the switch. To delete an SDP from the switch configuration, use the **no** form of the **service sdp vxlan** command. For example:

```
-> no service sdp 10
-> no service sdp 20
```

Verifying the SDP Configuration

To view the VXLAN SDP configuration for the switch, use the **show service sdp vxlan** command. For example:

```
-> show service sdp vxlan
Legend: (*) dyn unicast object
VxLAN SDP Info
                                                  Bind
SdpId       FarEnd/Group Addr Adm  Oper Intf      Count TTL  Description
-----------+----------------+----+----+---------+-----+----+-------------------
10          224.2.1.1        Up   Up   -          3     64   224.2.1.1 Group
20          10.10.10.2       Up   Up   1/1/1      2     64   To NodeB
30          10.10.10.3       Up   Up   1/1/2      1     64   To NodeC
```

Use the **far-end** and **multicast-group** parameter options with the **show service sdp vxlan** command to display only VXLAN SDPs associated with a specific far-end IP address or a specific multicast group IP address. For example:

```
-> show service sdp vxlan far-end 10.10.10.2
Legend: (*) dyn unicast object
VxLAN SDP Info
                                                  Bind
SdpId       FarEnd/Group Addr Adm  Oper Intf      Count TTL  Description
-----------+----------------+----+----+---------+-----+----+-------------------
20          10.10.10.2       Up   Up   1/1/1      2     64   To NodeB

Total SDPs: 1


-> show service sdp vxlan multicast-group 224.2.1.1
Legend: (*) dyn unicast object
VxLAN SDP Info
                                                  Bind
SdpId       FarEnd/Group Addr Adm  Oper Intf      Count TTL  Description
-----------+----------------+----+----+---------+-----+----+-------------------
10          224.2.1.1        Up   Up   -          3     64   224.2.1.1 Group

Total SDPs: 1
```

Binding VXLAN Services to SDPs

After an SDP is created, a VXLAN service is then bound to an SDP to direct one-way VXLAN encapsulated traffic to a single far-end node (unicast SDP) or to direct VXLAN encapsulated BUM traffic to all VTEPs that are members of the same multicast group (multicast SDP).

The multicast mode (tandem, head-end, or hybrid) configured for a VXLAN service determines the type of SDP to which the service can be bound. For example:

- A tandem mode VXLAN service can be bound only to one multicast SDP (an SDP with a multicast group address).

- A head-end mode VXLAN service can be bound to multiple unicast SDPs (an SDP with a far-end IP address).

- A hybrid mode VXLAN service can be bound to *both* of the following:
  - One multicast SDP for a group of member VTEPs.
  - Many unicast SDPs for VTEPs that do not participate in a multicast group.

Configuring an SDP Binding

Use the **service bind-sdp** command to bind a VXLAN service to an SDP. For example:

```
-> service 1 bind-sdp 10 description "Bind to PIM Group 224.2.1.1"
-> service 2 bind-sdp 20 description "Unicast Bind to 1.1.1.20 VTEP"
```

In these two examples, the switch will tunnel all traffic classified into the SAP associated with VXLAN service 1 through SDP 10 and will tunnel all traffic classified into the SAP associated with VXLAN service 2 through SDP 20.

To bind a range of services to a single SDP ID, use the **service bind-sdp** command with a hyphen to specify a range of services. For example:

```
-> service 1-100 bind-sdp 10 description "Binds Services 1-100 to PIM Group
225.2.1.1"
```

To bind a service to multiple SDP IDs, use the **service bind-sdp** command with a space between each service ID. For example:

```
-> service 3 bind-sdp 30 40 50 60 70 80 90 description "Binds Service 3 to
multiple SDP IDs"
```

When an SDP binding is created, the binding is automatically enabled. There is no method for configuring the administrative status for this type of service object.

Verifying the SDP Binding Configuration

To view the VXLAN SDP binding configuration for the switch, use the **show service bind-sdp** command with the **vxlan** parameter option. For example:

```
-> show service bind-sdp vxlan
Legend: * denotes a dynamic object
VxLAN Bind-SDP Info
SvcId    Bind-Sdp         Vnid      FarEnd/Group Addr    Oper Intf     Intf Name
--------+---------------+---------+-------------------+----+--------+------------
1        10:1             100       1.1.1.6              Up    -       Intf-101
1        20:1             100       1.1.1.2              Up    -       Intf-110
1        30:1             100       1.1.1.3              Up    -       Intf-101
1        100:1            100       224.1.1.100          Up    -
2        10:2             200       1.1.1.6              Up    -       Intf-101
2        20:2             200       1.1.1.2              Up    -       Intf-110
2        30:2             200       1.1.1.3              Up    -       Intf-101
2        200:2            200       224.1.1.200          Up    -

Total Bind-SDPs: 8
```

Use the **vnid** parameter option with the **show service bind-sdp** command to view only those bindings for a specific VXLAN Network ID. For example:

```
-> show service bind-sdp vnid 200
Legend: * denotes a dynamic object
VxLAN Bind-SDP Info
SvcId    Bind-Sdp       Vnid      FarEnd/Group Addr    Oper Intf    Intf Name
--------+---------------+--------+-------------------+----+--------+-----------
2        10:2           200       1.1.1.6              Up   -        Intf-101
2        20:2           200       1.1.1.2              Up   -        Intf-110
2        30:2           200       1.1.1.3              Up   -        Intf-101
2        200:2          200       224.1.1.200          Up   -

Total Bind-SDPs: 4
```

Configuring the UDP Port for a VXLAN Gateway

When the switch encapsulates a Layer 2 frame to create a VXLAN packet, a UDP header is added to the frame. This header contains a UDP destination port value, which by default is set to the well-known UDP port 4789. If it is necessary to change the UDP destination port number for VXLAN packets, consider the following guidelines:

- The UDP value is used in all encapsulated VXLAN packets on a gateway switch regardless of how many unique VXLAN segments are serviced by the switch. For example, if multiple VXLAN segments are serviced by the switch, all segments use the same UDP value for encapsulation even though each segment has a different VXLAN network ID.

- Only gateway devices using the same destination UDP port number can exchange encapsulated VXLAN frames.

- Avoid using the well-known UDP port numbers that are already reserved by IANA for other applications.

- Changing the UDP port number on the fly might stop the VXLAN traffic until all the other gateway devices in the network are configured with the same destination UDP port.

To change the UDP port value, use the **service vxlan udp-port** command. For example:

```
-> service vxlan udp-port 8472
```

To set the UDP port number back to the default value of 4789, use the **service vxlan udp-port** command with the **default** parameter option. For example:

```
-> service vxlan udp-port default
```

Use the **show service info** command to display the current value of the VXLAN UDP port number.

VXLAN Gateway Configuration Examples

This section contains the following OmniSwitch VXLAN gateway examples:

Example 1: Sample OmniSwitch VXLAN Topology

The following diagram provides a sample VXLAN overlay network topology to show how the VXLAN service framework works to basically extend Layer 2 traffic over a Layer 3 network:
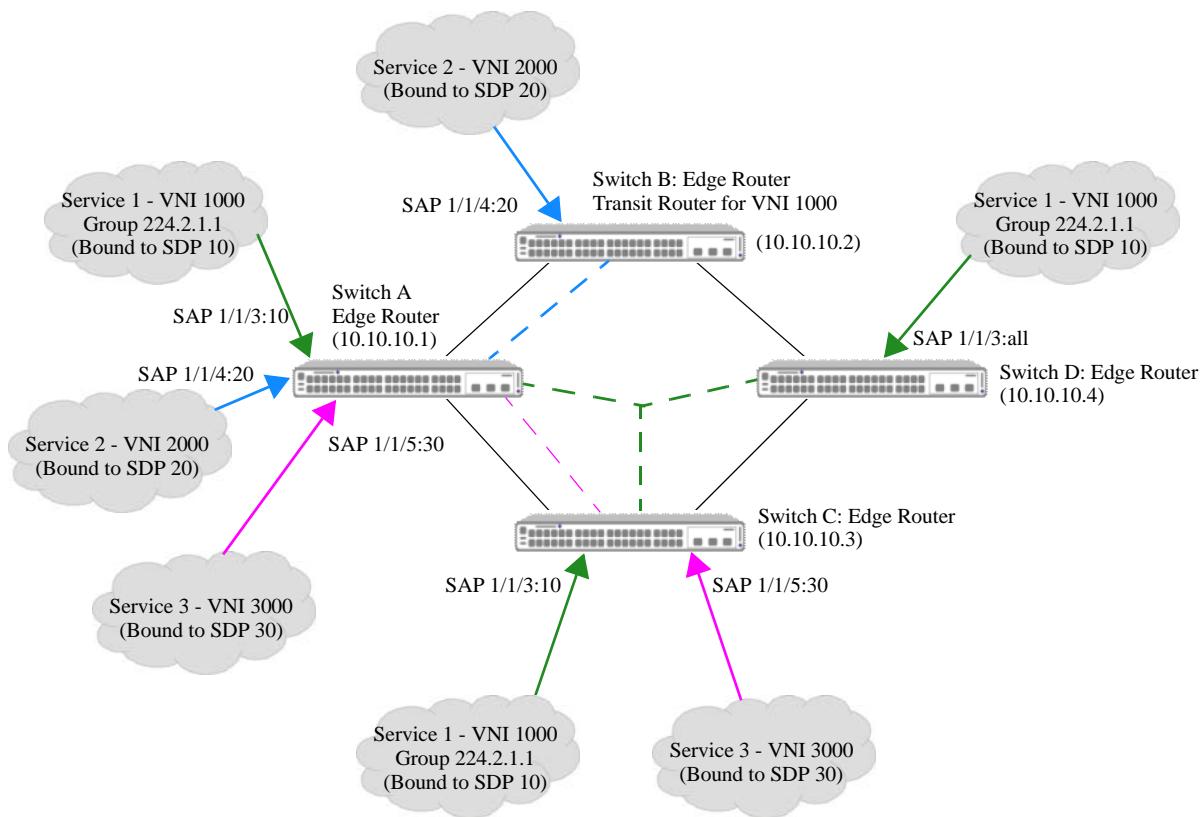
Figure 3: Sample VXLAN Configuration with Static Routes

In this sample topology:

- Switch A, Switch B, Switch C, and Switch D operate as OmniSwitch VXLAN gateways that serve as VXLAN Tunnel End Points (VTEPs) for the overlay network. Switch B also serves as a transit switch for service 1 (VNI 1000).

- Each switch is configured with static routes and BIDIR-PIM to handle the underlying routing framework.

- Service 1 is associated with VXLAN Network ID (VNI) 1000 and is bound to the SDP 10 tunnel interface, which participates in PIM group 224.2.1.1.

- Service 2 is associated with VNI 2000 and is bound to the SDP 20 tunnel interface, which is configured with a far-end IP address to carry unicast traffic.

- Service 3 is associated with VNI 3000 and is bound to the SDP 30 tunnel interface, which is also configured with a far-end IP address to carry unicast traffic.

The following CLI command examples are used to configure the sample VXLAN overlay network shown in :

Routing Configuration

The following commands are used to set up the underlying routing framework (Switch A setup is used for this example):

Standard VLANS

```
-> vlan 100 admin-state enable
-> vlan 100 members port 1/1/1 untagged
-> vlan 101 admin-state enable
-> vlan 101 members port 1/1/2 untagged
```

IP interfaces (including Loopback0 to provide the source IP address for the gateway)

```
-> ip interface "Loopback0" address 10.0.0.1
-> ip interface "v4lan100" address 10.100.0.1/16 vlan 100
-> ip interface "v4lan101" address 10.101.0.1/16 vlan 101
```

Static Routes to target VTEPs

```
-> ip static-route 10.10.10.2/32 gateway 10.100.0.2
-> ip static-route 10.10.10.4/32 gateway 10.100.0.2
-> ip static-route 10.200.0.0/16 gateway 10.100.0.2
-> ip static-route 10.10.10.3/32 gateway 10.101.0.3
-> ip static-route 10.201.0.0/16 gateway 10.101.0.3
```

BIDIR-PIM for multicast group 224.0.0.0

```
-> ip load pim
-> ip pim interface "v4lan100"
-> ip pim interface "v4lan101"
-> ip pim interface "Loopback0"
-> ip pim candidate-rp 10.10.10.1 224.0.0.0/4 bidir
-> ip pim cbsr 10.10.10.1
-> ip pim sparse admin-state enable
-> ip pim bidir admin-state enable
```

VXLAN Service Configuration

The following sets of commands provide examples of configuring the VXLAN service components for Switch A, Switch B, Switch C, and Switch D in the sample overlay network.

Switch A:

Service Access Ports

```
-> service access port 1/1/3
-> service access port 1/1/4
-> service access port 1/1/5
```

SDPs

```
-> service sdp 10 vxlan multicast-group 224.2.1.1 ttl 10 description "PIM Group
224.2.1.1"
-> service sdp 20 vxlan far-end 10.10.10.2 description "To Switch B"
-> service sdp 30 vxlan far-end 10.10.10.3 description "To Switch C"
```

VXLAN Services

```
-> service 1 vxlan vnid 1000 multicast-mode tandem stats enable description "VXLAN
Service for VNID 1000"
-> service 2 vxlan vnid 2000 multicast-mode head-end description "VXLAN Service for
VNID 2000"
-> service 3 vxlan vnid 3000 multicast-mode head-end description "VXLAN Service for
VNID 3000"
```

SAPs

```
-> service 1 sap port 1/1/3:10 stats enable
-> service 2 sap port 1/1/4:20 stats enable
-> service 3 sap port 1/1/5:30 stats enable
```

SDP Bindings

```
service 1 bind-sdp 10
service 2 bind-sdp 20
service 3 bind-sdp 30
```

Switch B:

Service Access Ports

```
-> service access port 1/1/4
```

SDPs

```
-> service sdp 20 vxlan far-end 10.10.10.1 description "To Switch A"
```

VXLAN Services

```
-> service 2 vxlan vnid 2000 multicast-mode head-end description "VXLAN Service for
VNID 2000"
```

SAPs

```
-> service 2 sap port 1/1/4:20 stats enable
```

SDP Bindings

```
-> service 2 bind-sdp 20
```

Switch C:

Service Access Ports

```
service access port 1/1/3
service access port 1/1/5
```

SDPs

```
-> service sdp 10 vxlan multicast-group 224.2.1.1 ttl 10 description "PIM Group
224.2.1.1"
-> service sdp 30 vxlan far-end 10.10.10.1 description "To Switch A"
```

<u>VXLAN Services</u>

```
-> service 1 vxlan vnid 1000 multicast-mode tandem description "VXLAN Service for
VNID 1000"
-> service 3 vxlan vnid 3000 multicast-mode head-end description "VXLAN Service for
VNID 3000"
```

<u>SAPs</u>

```
-> service 1 sap port 1/1/3:10
-> service 3 sap port 1/1/5:30
```

<u>SDP Bindings</u>

```
-> service 1 bind-sdp 10
-> service 3 bind-sdp 30
```

Switch D:

<u>Service Access Ports</u>

```
-> service access port 1/1/3
```

<u>SDPs</u>

```
-> service sdp 10 vxlan multicast-group 224.2.1.1 ttl 10 description "PIM Group
224.2.1.1"
```

<u>VXLAN Services</u>

```
-> service 1 vxlan vnid 1000 multicast-mode tandem description "VXLAN Service for
VNID 1000"
```

<u>SAPs</u>

```
-> service 1 sap port 1/1/3:all
```

<u>SDP Bindings</u>

```
-> service 1 bind-sdp 10
```

Example 2: Interoperability with Server VTEPs

The following diagram provides a sample VXLAN overlay network topology in which an OmniSwitch VXLAN gateway VTEP interacts with a Data Center server VTEP.
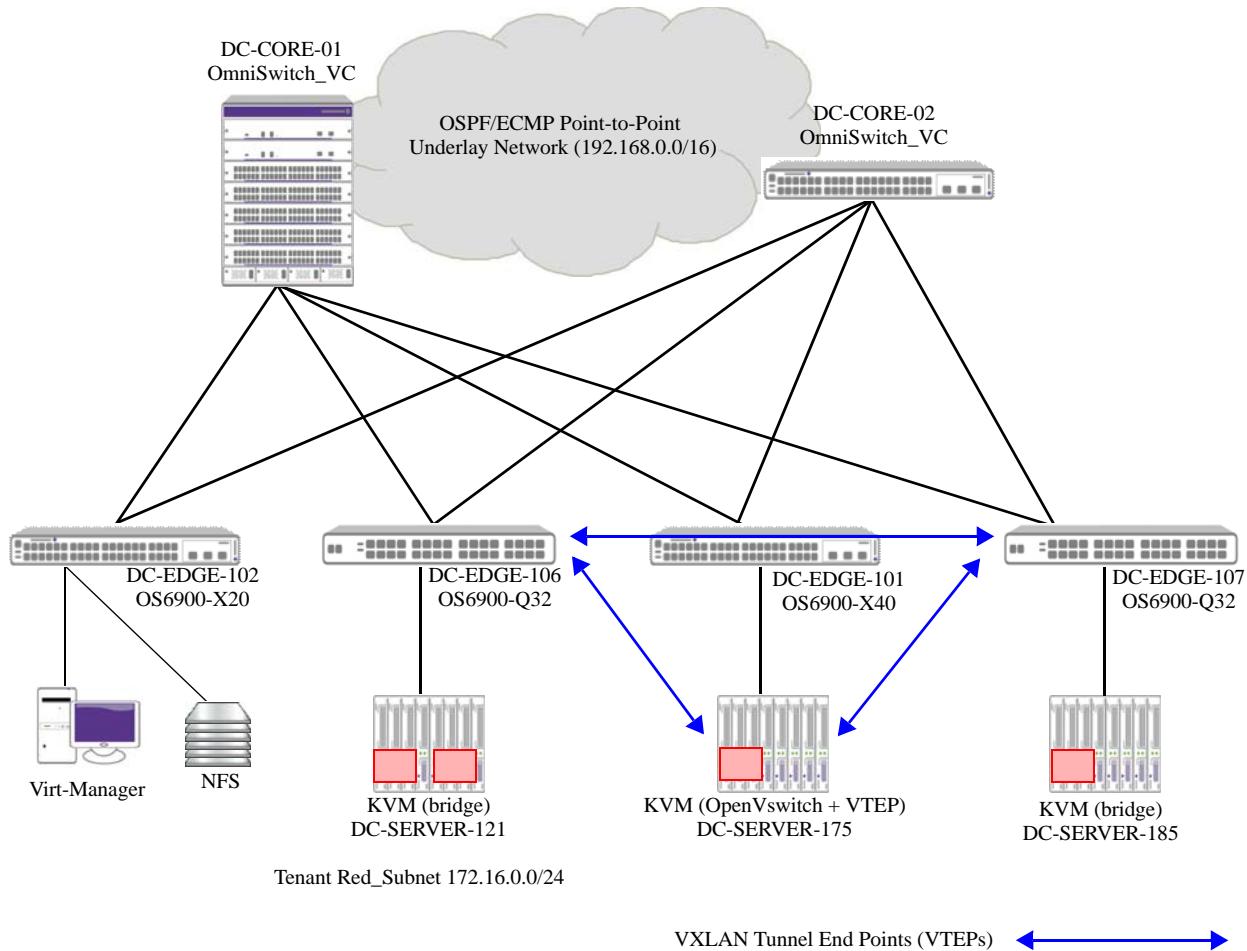
Figure 4: OmniSwitch VXLAN Gateway Interoperability

In this sample topology:

- Three Ubuntu/KVM Hypervisors are connected to different OmniSwitch edge switches.

- DC-SERVER-175 is connected to DC-EDGE-101 (an OS6900-X40) and has OpenVswitch installed with VXLAN support. A VXLAN Tunnel End Point (VTEP) is configured on this server.

- DC-SERVER-121 is connected to DC-EDGE-106 (an OS6900-Q32), which is configured as a VTEP to provide VXLAN gateway functionality.

- DC-SERVER-185 is connected to DC-EDGE-107 (an OS6900-Q32), which is also configured as a VTEP to provide VXLAN gateway functionality.

- The NFS Server and Virt-Manager devices (used for KVM management, guest instantiation) are connected to DC-EDGE-102 (an OS6900-X20).

- All the OmniSwitch edge switches are dual-homed to two core switches via link aggregate ports.

- The underlying network between the core switches is a point-to-point OSPF with ECMP and a range of 192.168.0.0/16. The use of OSPF with ECMP is specific to this example.

- VXLAN Snooping is configured on the core switches to detect and learn VXLAN traffic.

- Currently one Tenant Network (172.16.0.0/16) is configured. This network is referred to as the "tenant red_subnet".

- Virtual Machines (VMs) in the "tenant red_subnet" can communicate with each other through the VXLAN tunnels formed between the two OmniSwitch VXLAN gateway switches (DC-EDGE-106 and DC-EDGE-107) and the KVM VXLAN gateway server (DC-SERVER-175).

- VMs in the "tenant red_subnet" can also migrate from one server to another through the provided VXLAN tunnel.

VXLAN Configuration on KVM

```
user@DC-SERVER-175:~$ cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04.1 LTS"
user@DC-SERVER-175:~$

user@DC-SERVER-175:~$ uname -a
Linux DC-SERVER-175 3.13.0-32-generic #57-Ubuntu SMP Tue Jul 15 03:51:08 UTC 2014
x86_64 x86_64 x86_64 GNU/Linux
```

Common OVS Commands

```
sudo ovs-vsctl show

sudo ovs-vsctl add-br blue_subnet - Adding a bridge

sudo ovs-vsctl add-port red_subnet vx107 - Set interface vx107 type=vxlan

options:remote_ip=100.0.0.107 options:key=5 options:dst_port=4789 - adding a VXLAN
interface

user@DC-SERVER-175:~$ sudo ovs-vsctl -V
ovs-vsctl (Open vSwitch) 2.0.2
Compiled Aug 15 2014 14:31:02
user@DC-SERVER-175:~$

user@DC-SERVER-175:~$ virsh list

 Id    Name                          State
----------------------------------------------------
 4     red_vm_10                     running
 5     red_vm_11                     running

user@DC-SERVER-175:~$ sudo ovs-vsctl show

[sudo] password for user:
5d9b144b-8575-4b2d-99fc-af0c1f73f6f3
    Bridge mybridge
        Port "eth4"
            Interface "eth4"
```

```
        Port mybridge
            Interface mybridge
                type: internal
    Bridge blue_subnet
        Port blue_subnet
            Interface blue_subnet
                type: internal
    Bridge red_subnet
        Port "vnet0"
            Interface "vnet0"
        Port "virbr1"
            Interface "virbr1"
        Port "vx2"
            Interface "vx2"
                type: vxlan
                options: {dst_port="4789", key="5", remote_ip="100.0.0.106"}
        Port red_subnet
            Interface red_subnet
                type: internal
        Port "vnet1"
            Interface "vnet1"
        Port "vx107"
            Interface "vx107"
                type: vxlan
                options: {dst_port="4789", key="5", remote_ip="100.0.0.107"}
        Port "virbr0"
            Interface "virbr0"
    ovs_version: "2.0.2"
user@DC-SERVER-175:~$

user@DC-SERVER-175:~$ netstat -nr
Kernel IP routing table
Destination      Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0          10.255.92.254   0.0.0.0         UG        0 0          0 eth0
10.255.92.0      0.0.0.0         255.255.255.0   U         0 0          0 eth0
100.0.0.0        192.168.17.1    255.255.255.0   UG        0 0          0 mybridge
172.16.0.0       0.0.0.0         255.255.255.0   U         0 0          0 red_subnet
192.168.0.0      192.168.17.1    255.255.0.0     UG        0 0          0 mybridge
192.168.17.0     0.0.0.0         255.255.255.0   U         0 0          0 mybridge

user@DC-SERVER-175:~$ df
Filesystem                     1K-blocks      Used Available Use% Mounted on
/dev/mapper/dc--server2--vg-root 472044848  2770872 445272440   1% /
none                                   4        0         4   0% /sys/fs/cgroup
udev                             4073816        4   4073812   1% /dev
tmpfs                             823608      820    822788   1% /run
none                                5120        0      5120   0% /run/lock
none                             4118024        0   4118024   0% /run/shm
none                              102400        0    102400   0% /run/user
/dev/sda1                         240972    37050    191481  17% /boot
192.168.25.123:/export/users    94558208 25460736  64271360  29% /var/lib/
libvirt/images/DC-VM-NFS
user@DC-SERVER-175:~$

user@DC-SERVER-175:~$

user@DC-SERVER-175:~$ ifconfig
```

```
blue_subnet Link encap:Ethernet  HWaddr 7a:2a:49:7a:3f:47
            inet6 addr: fe80::d8e2:8aff:fed4:c260/64 Scope:Link
            UP BROADCAST RUNNING  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:0 (0.0 B)  TX bytes:648 (648.0 B)

eth0        Link encap:Ethernet  HWaddr 00:22:19:6d:a4:72
            inet addr:10.255.92.175  Bcast:10.255.92.255  Mask:255.255.255.0
            inet6 addr: fe80::222:19ff:fe6d:a472/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:10231779 errors:0 dropped:0 overruns:0 frame:0
            TX packets:6488820 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:1111118213 (1.1 GB)  TX bytes:2980739659 (2.9 GB)

eth4        Link encap:Ethernet  HWaddr 00:00:c9:e3:be:16
            inet6 addr: fe80::200:c9ff:fee3:be16/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:906707 errors:0 dropped:0 overruns:76 frame:0
            TX packets:279029 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:997038377 (997.0 MB)  TX bytes:90927732 (90.9 MB)

lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:104094 errors:0 dropped:0 overruns:0 frame:0
            TX packets:104094 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:18718499 (18.7 MB)  TX bytes:18718499 (18.7 MB)

mybridge    Link encap:Ethernet  HWaddr 00:00:c9:e3:be:16
            inet addr:192.168.17.5  Bcast:192.168.17.255  Mask:255.255.255.0
            inet6 addr: fe80::28c6:adff:feed:6e2e/64 Scope:Link
            UP BROADCAST RUNNING  MTU:1500  Metric:1
            RX packets:264144 errors:0 dropped:0 overruns:0 frame:0
            TX packets:240355 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:952785623 (952.7 MB)  TX bytes:90901898 (90.9 MB)

red_subnet Link encap:Ethernet  HWaddr 46:91:54:dc:6e:41
            inet addr:172.16.0.175  Bcast:172.16.0.255  Mask:255.255.255.0
            inet6 addr: fe80::d8c4:8aff:fece:f06/64 Scope:Link
            UP BROADCAST RUNNING  MTU:1500  Metric:1
            RX packets:45374 errors:0 dropped:10 overruns:0 frame:0
            TX packets:25238 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:3338732 (3.3 MB)  TX bytes:2427988 (2.4 MB)

virbr1      Link encap:Ethernet  HWaddr 52:54:00:cf:55:47
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

```
vnet0     Link encap:Ethernet  HWaddr fe:54:00:5e:25:0d
          inet6 addr: fe80::fc54:ff:fe5e:250d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:29111 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30431 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:2797702 (2.7 MB)  TX bytes:2916490 (2.9 MB)

vnet1     Link encap:Ethernet  HWaddr fe:54:00:f2:0d:10
          inet6 addr: fe80::fc54:ff:fef2:d10/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:29653 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30972 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:2848802 (2.8 MB)  TX bytes:2968216 (2.9 MB)

user@DC-SERVER-175:~$
```

Additional Configuration Notes

Virt-Manager does not directly interact with OVS (it works on a Linux Bridge). When a subnet is created with Virt-Manager, the **VM.xml** file will have to be edited to change a few settings. The following link provides more information about how to do this:

```
http://www.opencloudblog.com/?p=177
```

The other alternative is to downgrade to an earlier version of OpenVswitch and use a bridge-compatibility module. The following link provides information about creating bridges and tunnels:

```
http://networkstatic.net/configuring-vxlan-and-gre-tunnels-on-openvswitch/
```

OmniSwitch Configuration

```
DC-EDGE-106-> show configuration snapshot svcmgr
! SVCMGR:
service access port 1/1/17A
service access port 1/1/17C
service access port 1/1/18A
service access port 1/1/19B
service sdp 106 vxlan multicast-group 239.0.0.0
service sdp 175 vxlan far-end 192.168.17.5 description "kvm_175"
service sdp 185 vxlan far-end 100.0.0.107 description "AOS_DC_EDGE_107"
service 5 vxlan vnid 5 multicast-mode head-end vlan-xlation enable
service 10 vxlan vnid 10 multicast-mode tandem
service 5 sap port 1/1/17A:0
service 5 sap port 1/1/19B:0
service 10 sap port 1/1/17A:10
service 5 bind-sdp 175 185
service 10 bind-sdp 106
```

```
DC-EDGE-106-> show ip ospf neighbor
  IP Address        Area Id          Router Id       Vlan   State   Type
---------------+---------------+---------------+------+-------+--------
192.168.254.20   0.0.0.0          100.0.0.96       1020    Full   Dynamic
192.168.254.22   0.0.0.0          100.0.0.95       1022    Full   Dynamic


DC-EDGE-106-> show ip router database dest 192.168.17.0/24
Legend: + indicates routes in-use
        b indicates BFD-enabled static route
        i indicates interface static route
        r indicates recursive static route, with following address in brackets

  Destination       Gateway          Interface       Protocol Metric  Tag   Misc-Info
-------------------+---------------+--------------+--------+-------+-----+---------
+  192.168.17.0/24  192.168.254.20  p2p_net1020    OSPF          1     0
+  192.168.17.0/24  192.168.254.22  p2p_net1022    OSPF          1     0


DC-EDGE-106-> show service 5
VxLAN Service Detailed Info
  Service Id      : 5,                  Description      : ,
  VNID            : 5 (0.0.5),
  Multicast-Mode  : Headend,
  Admin Status    : Up,                 Oper Status      : Up,
  Stats Status    : No,                 Vlan Translation : Yes,
  Service Type    : VxLAN,              Allocation Type  : Static,
  MTU             : 9194,               Def Mesh VC Id   : 5,
  SAP Count       : 2,                  SDP Bind Count   : 2,
  Ingress Pkts    : 0,                  Ingress Bytes    : 0,
  Egress Pkts     : 0,                  Egress Bytes     : 0,
  Mgmt Change     : 03/13/2014 16:24:43, Status Change   : 03/13/2014 16:26:18


DC-EDGE-106-> show service 5 debug-info
Legend: * denotes a dynamic object
VxLAN Service 5 Debug Info
  Admin : Up,        Oper : Up,     Stats      : N,        VlanXlation : Y,
  VNID  : 5 (0.0.5),                MCast-Mode : Headend,
  VFI   : 1,         McIdx : 3,     StatsHandle: 0

Sap Trusted:Priority/    Sap Description/   Stats/
Identifier    Adm  Oper Stats Sdp FarEnd Addr   Intf  Sdp Intf Name  VP  L2 McIdx
-------------+----+----+-----+----------------+--------+-------------+---+-------
sap:1/1/17A:0 Up   Down  N          Y:x         1/1/17A    -          1   0
sap:1/1/19B:0 Up   Up    N          Y:x         1/1/19B    -          2   0
sdp:175:5     Up   Up    Y   192.168.17.5    -      p2p_net1022    5   1
sdp:185:5     Up   Up    Y   100.0.0.107     -      p2p_net1022    6   1

Total Ports: 4
```

```
DC-EDGE-106-> show mac-learning domain vxlan
Legend: Mac Address: * = address not valid,

        Mac Address: & = duplicate static address,

Domain  Vlan/SrvcId[ISId/vnId] Mac Address       Type     Operation  Interface
-------+--------------------+----------------+--------+---------+-------------
 VXLAN                  5:5  00:0e:1e:0c:58:94  dynamic  servicing   sap:1/1/19B
 VXLAN                  5:5  00:0e:1e:0c:58:97  dynamic  servicing   sap:1/1/19B
 VXLAN                  5:5  52:54:00:3e:40:1d  dynamic  servicing   sap:1/1/19B
 VXLAN                  5:5  46:91:54:dc:6e:41  dynamic  servicing   sdp:175:5
 VXLAN                  5:5  52:54:00:5e:25:0d  dynamic  servicing   sdp:175:5

 VXLAN                  5:5  52:54:00:f2:0d:10  dynamic  servicing   sdp:175:5
 VXLAN                  5:5  00:0e:1e:06:87:8c  dynamic  servicing   sdp:185:5

 Total number of Valid MAC addresses above = 7
```

Example 3: Sample VXLAN Topology Without Routing Protocols

When VXLAN is used in a network topology that does not utilize routing protocols (such as OSPF or static routes), multiple direct routes on the same VXLAN interface is not supported. Only one NH (nexthop) per interface is supported. In this scenario, configuring direct connections between the VTEP switches is highly recommended. In addition, make sure that any VTEP ARP is resolved before traffic is sent on the SAP port.

The following diagram provides a sample VXLAN overlay network topology in which there is no routing protocol configuration:
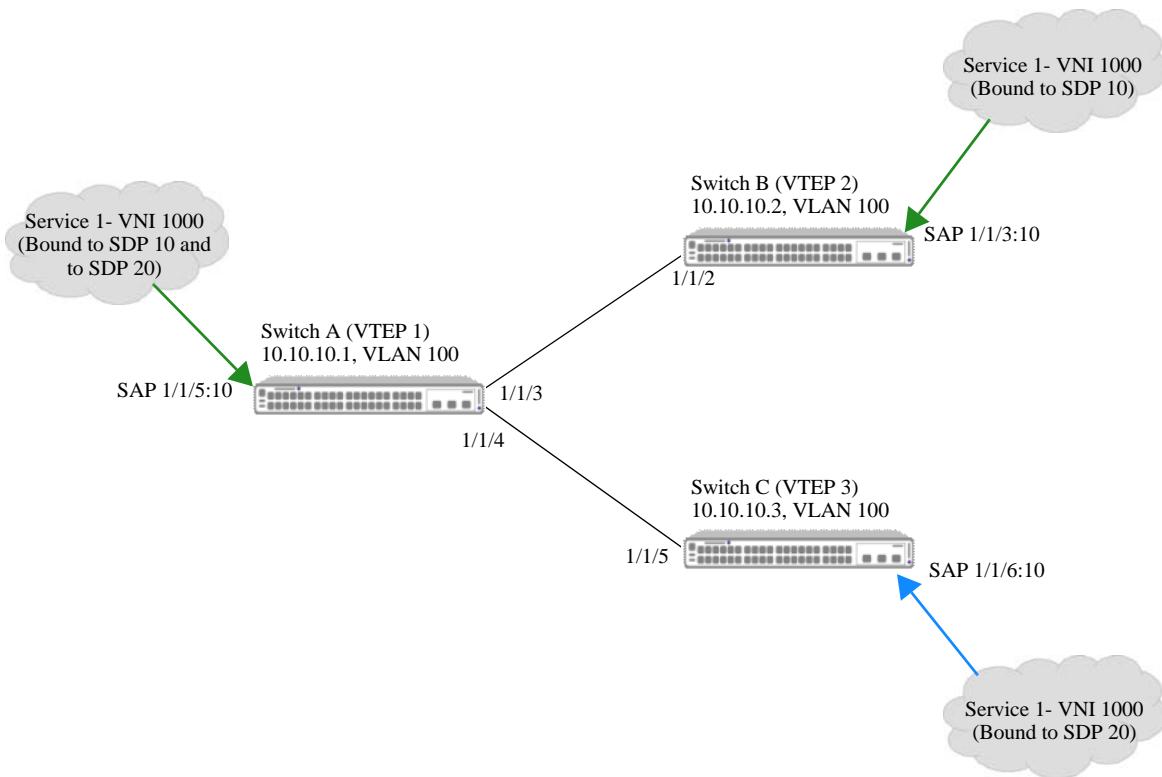
Figure 5: Sample VXLAN Configuration with no Routing Framework

In this sample topology:

* Switch A, Switch B, and Switch C operate as OmniSwitch VXLAN gateways that serve as VXLAN Tunnel End Points (VTEPs) for the overlay network.

* There is no underlying routing framework. The same VLAN ID and subnet is used on all three VTEPs.

* Switch A is directly connected to Switch B and Switch C through the same VLAN; there is no transit switch between these connections.

* Service 1 is associated with VXLAN Network ID (VNI) 1000 and is bound to the SDP 10 and SDP 20 tunnel interfaces, which are configured with a far-end IP address to carry unicast traffic.

CLI Configuration

The following CLI command examples are used to configure the sample VXLAN overlay network shown in :

Switch A:

```
Standard VLAN and Port Assignment
-> vlan 100 admin-state enable
-> vlan 100 members port 1/1/3 untagged
-> vlan 100 members port 1/1/4 untagged


IP Interfaces (including the Loopback0 source IP address for the gateway)
-> ip interface "Loopback0" address 10.0.0.1
-> ip interface "v4lan100" address 10.100.0.1/16 vlan 100


Service Access Ports
-> service access port 1/1/5


SDPs
-> service sdp 10 vxlan far-end 10.10.10.2 description "To Switch B"
-> service sdp 20 vxlan far-end 10.10.10.3 description "To Switch C"


VXLAN Services
-> service 1 vxlan vnid 1000 multicast-mode head-end description "VXLAN Service for
VNID 1000"


SAPs
-> service 1 sap port 1/1/5:10 stats enable


SDP Bindings
-> service 1 bind-sdp 10 20
```

Switch B:

```
Standard VLAN and Port Assignment
-> vlan 100 admin-state enable
-> vlan 100 members port 1/1/2 untagged


IP Interfaces (including the Loopback0 source IP address for the gateway)
-> ip interface "Loopback0" address 10.0.0.2
-> ip interface "v4lan100" address 10.100.0.2/16 vlan 100


Service Access Ports
-> service access port 1/1/3


SDPs
-> service sdp 10 vxlan far-end 10.10.10.1 description "To Switch A"


VXLAN Services
-> service 1 vxlan vnid 1000 multicast-mode head-end description "VXLAN Service for
VNID 1000"
```

<u>SAPs</u>
```
-> service 1 sap port 1/1/3:10
```

<u>SDP Bindings</u>
```
-> service 1 bind-sdp 10
```

Switch C:

<u>Standard VLAN and Port Assignment</u>
```
-> vlan 100 admin-state enable
-> vlan 100 members port 1/1/5 untagged
```

<u>IP Interfaces (including the Loopback0 source IP address for the gateway)</u>
```
-> ip interface "Loopback0" address 10.0.0.3
-> ip interface "v4lan100" address 10.100.0.3/16 vlan 100
```

<u>Service Access Ports</u>
```
-> service access port 1/1/6
```

<u>SDPs</u>
```
-> service sdp 20 vxlan far-end 10.10.10.1 description "To Switch A"
```

<u>VXLAN Services</u>
```
-> service 1 vxlan vnid 1000 multicast-mode head-end description "VXLAN Service for
VNID 1000"
```

<u>SAPs</u>
```
-> service 1 sap port 1/1/6:10
```

<u>SDP Bindings</u>
```
-> service 1 bind-sdp 20
```

Verifying the VXLAN Configuration

Displaying the VXLAN configuration is helpful to verify the actual configuration on each VXLAN gateway switch in the topology and to troubleshoot VXLAN service connectivity. To display information about VXLAN components, use the **show** commands listed in this section.

show service access	Displays the service access port configuration.
show service l2profile	Displays the Layer 2 profile definitions. These profiles are applied to service access ports to determine how Layer 2 control protocol frames are processed on these ports.
show service	Displays the service configuration.
show service ports	Displays all the virtual ports (SAPs, SDPs) that are associated with a VXLAN service.
show service spb sap	Displays the configuration information for the specified SAP ID associated with the specified service.
show service sdp vxlan	Displays the VXLAN SDP configuration.
show service bind-sdp vxlan	Displays the VXLAN service-to-SDP binding configuration.
show service debug-info	Displays debug information for the virtual ports associated with a service.
show service info	Displays the Service Manager configuration for the local switch, which includes VXLAN information.
show service counters	Displays the traffic statistics for the specified service and associated virtual ports.

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

5 Configuring VXLAN Snooping

The OmniSwitch Virtual eXtensible LAN (VXLAN) Snooping feature attempts to detect and identify VXLAN traffic by sampling packets to determine if they are VXLAN encapsulated packets. Once this type of traffic is identified, VXLAN Snooping collects and stores information about the flow in a database on the local switch. Additional configurable options for this feature include the ability to apply QoS policy list rules to the identified flow and SNMP trap generation.

In data centers that are using the VXLAN overlay technology, servers running VXLAN look like individual machines talking IP with other machines. Network devices do not have any visibility into the overlay network, which can lead to the following:

- Difficulty analyzing errors and cross correlating device and network location.

- The network can only act on the QoS of the external packet, not on the encapsulated Ethernet frame within the packet. For example, the QoS markings of the inner frame have to be trusted and there is no way to differentiate between VXLAN Network IDs (VNIs) and individual devices, such as Virtual Machines (VMs).

The OmniSwitch implementation of VXLAN Snooping makes use of the well-known VXLAN encapsulation format to identify, analyze, and optionally apply QoS policies to VXLAN packets based on a VNI and the inner frame fields. Information obtained from this process is stored and used to track the VMs and VNIs discovered in the network.

Some of the key benefits of using VXLAN Snooping include the following:

- Functionality is enabled or disabled on a per-port basis.

- A database of discovered devices and their respective VNIs along with statistics.

- Two filtering modes for policy lookup to accommodate granularity and scalability of switch resources.

- QoS policies offer a wide range of control.

- SNMP trap generation for VXLAN packet flow learning, aging out, and resource limits.

- Can be used on UNP ports for dynamic assignment and applying QoS policy lists.

Note. Throughout this chapter, the terms "VXLAN Snooping" and "VM Snooping" are interchangeable.

In This Chapter

This chapter provides an overview of the VXLAN Snooping feature and describes how to configure the port-based functionality through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following information and procedures are included in this chapter:

- "VXLAN Snooping Defaults" on page 5-3.

- "Quick Steps for Configuring VXLAN Snooping" on page 5-4.

- "VXLAN Snooping Overview" on page 5-6.

- "Interaction With Other Features" on page 5-10.

- "Configuring VXLAN Snooping" on page 5-12.

- "VXLAN Snooping Configuration Example" on page 5-19

- "Verifying the VXLAN Snooping Configuration" on page 5-21.

For more information about VXLAN, see Chapter 4, "Configuring a VXLAN Gateway."

VXLAN Snooping Defaults

By default, the VXLAN Snooping functionality is disabled on the switch. When VXLAN Snooping is enabled, the following global default values are applied:

Parameter Description	Command	Default
VXLAN Snooping policy lookup mode	**vm-snooping policy-mode**	Basic, tagged and untagged inner frame header, default resources.
The status of trap generation	**vm-snooping trap**	Disabled
Trap threshold value to indicate that a percentage of filtering resources has been used.	**vm-snooping filtering-resource trap threshold**	80%
The number of packets-per-second that are sampled on VXLAN Snooping ports.	**vm-snooping sampling-rate**	1000 pps
The amount of time the switch waits before aging out learned VXLAN packet flows.	**vm-snooping aging-timer**	300 seconds
The UDP destination port for VXLAN frames.	**vm-snooping vxlan udp-port**	4789
The number of static policy rules configured for VXLAN Snooping	**vm-snooping static-policy rule**	None
The number of VXLAN flows and statistics logged to a local .csv file on the switch.	**vm-snooping logging-threshold**	5000
The status of VXLAN Snooping on a switch port or link aggregate.	**vm-snooping port**	Disabled

Quick Steps for Configuring VXLAN Snooping

The following quick steps provide a brief tutorial for configuring VXLAN Snooping to sample VXLAN packets on ports enabled for VXLAN Snooping:

1 Use the **vm-snooping admin-state** command to globally enable the VXLAN Snooping functionality on the switch:

```
-> vm-snooping admin-state enable
```

When VXLAN Snooping is globally enabled for the switch, the default values provided in "VXLAN Snooping Defaults" on page 5-3 are applied.

2 Use the **vm-snooping port** command to enable VXLAN Snooping functionality on one or more switch ports or link aggregates:

```
-> vm-snooping port 1/10-25 admin-state enable
-> vm-snooping linkagg 5-10 admin-state enable
```

Once VXLAN Snooping is globally enabled for the switch and enabled on ports or link aggregates, the switch starts to sample packets received on the ports to detect and identify VXLAN packet flows.

3 *Optional*. Use the **vm-snooping trap** command to enable trap generation when a VXLAN packet flow is learned or ages out, system resources reach a user-specified threshold, or a module or port on which VXLAN flows were learned goes down. For example:

```
-> vm-snooping trap enable
```

Note. *Optional*. Verify the VXLAN Snooping configuration using the **show vm-snooping config** and **show vm-snooping port** commands. For example:

```
-> show vm-snooping config
VM-Snooping Status                        : Enable,
Trap                                      : Disable,
Trap-Threshold                            : 80,
Policy-Mode                               : Basic
Policy-Resource                           : Default
VM Inner Header                           : Tagged and Untagged,
Aging-Timer                               : 300 seconds,
UDP-Port(s)                               : 4789,
Sampling-Rate                             : 1000
Logging-Threshold                         : 5000
Qos Allocation Status                     : Success

-> show vm-snooping port
Port    VM-Snooping    vNP
-------+-----------+--------
1/2/1   Enable        Yes
1/2/2   Enable        No
1/2/2   Enable        Yes
1/2/2   Enable        Yes
0/12    Disable       No
```

To display VXLAN packet flows already learned on the switch, use the **show vm-snooping database** command. For example:

```
-> show vm-snooping database
Total number of VM Flows: 15

         VXLAN                    VM            VM          VM
Port     PORT   VNI       SRC MAC           VLAN      SRC IP
------+------+--------+----------------+-----+-------------------------------
1/1/3  4789   2000      00:12:01:00:00:01     -  200.200.200.1
1/1/3  4789   2001      00:12:01:00:00:02     -  200.200.201.1
1/1/3  4789   2002      00:12:01:00:00:03     -  200.200.202.1
1/1/3  4789   2003      00:12:01:00:00:04     -  200.200.203.1
1/1/3  4789   2004      00:12:01:00:00:05     -  200.200.204.1
1/1/3  4789   1234      00:00:04:00:00:00     -  2.0.0.0
1/1/3  4789   1234      00:00:04:00:00:01     -  2.0.0.1
1/1/3  4789   1234      00:00:04:00:00:02     -  2.0.0.2
1/1/3  4789   1234      00:00:04:00:00:03     -  2.0.0.3
1/1/3  4789   1234      00:00:04:00:00:04     -  2.0.0.4
1/2/2  4789   43        00:00:00:88:00:01 8      88.0.0.1
1/2/2  4789   43        00:00:00:88:00:02 8      88.0.0.2
1/2/2  4789   43        00:00:00:88:00:03 8      88.0.0.3
1/2/2  4789   43        00:00:00:88:00:04 8      88.0.0.4
1/2/2  4789   43        00:00:00:88:00:05 8      88.0.0.5
```

See the *OmniSwitch AOS Release 8 CLI Reference Guide* for information about the fields in this display.

VXLAN Snooping Overview

The OmniSwitch VXLAN Snooping feature attempts to detect and identify VXLAN packets by scanning IP packets received on VXLAN Snooping ports. This type of functionality gives an administrator:

- Visibility into the VXLAN segments and the devices that are active within each segment, such as Virtual Machines (VMs).

- The ability to apply QoS policies to VXLAN encapsulated packets based on the contents of header fields and the fields of the inner Ethernet frame.

VXLAN Snooping is a software task that analyzes the VXLAN packets that are sampled by the hardware in order to record the packet information and determine if any QoS should be applied. This functionality provides visibility, traceability, and control for Layer 2 virtual overlay networks by identifying the following parameters of VXLAN encapsulated packets:

- Inner source MAC address (the source MAC address of the original Ethernet frame that was encapsulated).

- Inner source IPv4, IPv6 address (the source IP address of the original Ethernet frame that was encapsulated).

- Outer UDP destination port used by the VXLAN Tunnel End Point (VTEP).

- VXLAN Network Identifier (VNI).

- Layer 4 destination and source port.

- IP protocol version.

VXLAN Snooping is enabled globally on the switch and on a per-port basis. The overlay network information needed and where QoS should be enforced determines on which ports this functionality should be enabled. For example, in a data center topology enable VXLAN Snooping as follows:

- On server-facing ports to learn more detailed information about VMs, such as their network location.

- On a server-facing port to enforce QoS for a specific VM.

- On all ports to determine the path that VMs are following.

Enabling VXLAN Snooping on a port or link aggregate triggers the sampling of IP packets on that port or aggregate. The following diagram provides a high-level example of the VXLAN Snooping process:
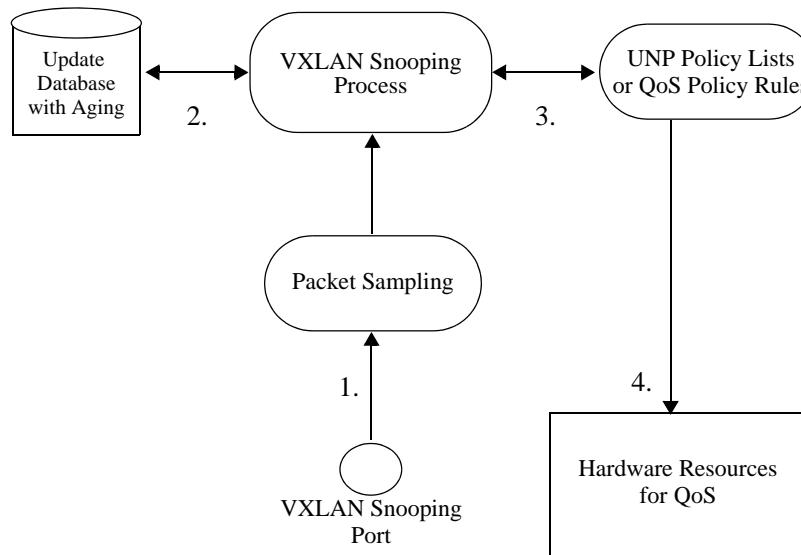
1 Packets arrive on a VXLAN Snooping port, which triggers random packet sampling. If VXLAN Snooping is not enabled on the port, the packet passes through the switch for standard processing.

2 If a VXLAN packet is detected and the packet information is not known to the switch, parameters identifying the packet are logged into the VXLAN Snooping database on the local switch. An aging timer is applied to each database entry.

3 QoS policies are checked to see if they match the VXLAN packet.

4 If a VXLAN packet matches a QoS policy rule, hardware resources are allocated for the matching rule and QoS is applied to the packet flow. If not, the learned VXLAN packet is forwarded on to the intended destination.

QoS for VXLAN Packet Flows

By default, the VXLAN Snooping function does not apply any QoS to the discovered VXLAN packet flows. However, the following methods for applying QoS are available:

• QoS applied through standard policies. These policies apply only to the outer header information of the encapsulated VXLAN packet.

• QoS applied through standard policies that contain a VXLAN condition. This type of policy is used to apply QoS to the inner frame information of the encapsulated packet.

• QoS through Universal Network Profile (UNP) policy lists. When a VXLAN packet flow is classified into a UNP, any policy list associated with that UNP is applied to the packet flow. A policy list may contain VXLAN conditions or standard policy conditions.

VXLAN Snooping Policy Modes

There are two types of VXLAN Snooping policy lookup modes and resource reservation: basic and advanced. The policy mode determines how QoS resources are allocated for VXLAN Snooping policies.

- When VXLAN Snooping is operating in the basic policy mode, a VXLAN QoS policy condition can classify on the following fields:
  - VNI (minimum required)
  - UDP port
  - Inner source MAC address
  - Inner source IPv4 address

- When VXLAN Snooping is operating in the advanced policy mode, a VXLAN QoS policy condition can classify on the following fields:
  - VNI (minimum required)
  - UDP port
  - Inner source MAC address
  - Inner IP protocol version
  - Inner source IPv6 address or IPv4 address
  - Inner destination and/or source port (TCP/UDP ports)

QOS policy rules support policy conditions that are applied to the inner frame of a VXLAN packet. Such policy conditions are identified through the use of the **vxlan** keyword. For example, the following rule contains a VXLAN policy condition:

```
-> policy condition c1 vxlan vni 1234 udp port 4789
-> policy condition c1 vxlan inner source mac 00:11:22:33:44:00
-> policy condition c1 vxlan inner source ip 10.10.10.10
-> policy action a1 disposition dscp 45
-> policy rule r1 condition c1 action a1
```

When a VXLAN condition is included in a QoS policy rule, the hardware resources for that rule are not used until a VXLAN packet flow matches that rule. The advantage to this is that switch resources reserved for VXLAN Snooping are not used until needed. This conserves the usage of switch resources that were allocated for this feature.

Static Policy Rules

VXLAN Snooping policies are dynamically installed in the hardware when a VXLAN packet flow triggers the application of the policy. However, there is a method for manually allocating the necessary resources before a packet flow matches the rule conditions. This is accomplished by configuring a static VXLAN Snooping policy rule.

When a static policy rule is configured, the rule is installed in the hardware if VXLAN Snooping is globally enabled. When traffic is received on a VXLAN Snooping port that matches a static policy condition, the static policy action is applied.

By default, policies are dynamically installed in the hardware when the VXLAN traffic is sampled to the CPU and after a lookup in the pre-configured VXLAN policy conditions. If a match is found, the policies are installed in the hardware.

Both static and dynamic policies are installed in the order in which the policy rules were configured in the QoS policy database. Rules that are members of a Universal Network Profile (UNP) policy list take precedence.

UNP

The Universal Network Profile (UNP) functionality is also configurable on a VXLAN Snooping port. When both are enabled on the same port, the following process is triggered:

1 UNP attempts to classify the ingress traffic based on the outer VXLAN packet information.

2 If the traffic is assigned to a UNP, the switch then checks if the UNP is associated with a QoS policy list.

3 If the UNP is associated with a QoS policy list, then that list is applied to the VXLAN packet flow.

4 If there is no matching UNP or the UNP does not use a QoS policy list, the switch checks if there are any QoS policies associated with the system default policy list that match the VXLAN flow parameters. If so, the default policy list rules are applied to the flow. If not, then no QoS is applied to the VXLAN packet flow.

UNP policy list rules take precedence over default policy list rules. The order in which policy rules within a specific list are applied is based on the precedence value assigned to the rule.

The following is an example configuration for a UNP QoS policy list:

```
-> policy condition c1 vxlan vni 1234 udp port 4789
-> policy condition c1 vxlan inner source mac 00:11:22:33:44:00
-> policy condition c1 vxlan inner source ip 10.10.10.10
-> policy action a1 disposition dscp 45
-> policy rule r1 condition c1 action a1 no default-list
-> policy list l1 type unp
-> policy list l1 rule r1
```

VXLAN Snooping Database

The VXLAN Snooping database contains a list of learned VXLAN packet flows. In addition, a packet counter for each VXLAN flow learned on a VXLAN Snooping port is kept for statistics generation. The database entries and statistics are displayed using the **show vm-snooping database** command and the **show vm-snooping statistics** command (see "Verifying the VXLAN Snooping Configuration" on page 5-21).

Each database entry is subject to a configurable aging period. At the time a VXLAN packet flow is learned, the configuration time for the flow is also saved. If there are no VXLAN packets detected for this same flow during the specified aging time period, the learned packet flow is removed from the VXLAN database. In addition, the hardware resources allocated for any VXLAN policies associated with the flow become available for other VXLAN policies.

Logging Database Entries

By default, a logging function is enabled when VXLAN Snooping is enabled for the switch. This function logs the database entries and statistics gathered to the **vm_snoop_db_flow_rec.csv** file and the **vm_snoop_hw_stats_rec.csv** file in the **/flash/switch/bridge/vm_snoop/** directory on the local switch.

Logging database entries to a local file on the switch helps to maintain a history of VXLAN packet flows on the switch. In addition, the information in these files is also accessed by Alcatel-Lucent Enterprise network management tools to provide management and visibility of the overlay network traffic.

The number of entries logged is a configurable value, and it is also possible to turn the logging function off. However, turning logging off does not stop the learning and recording of VXLAN packet flow entries into the VXLAN Snooping software database. For more information, see "Configuring Database Entry Logging" on page 5-17.

Interaction With Other Features

This section contains important information about how VXLAN Snooping functionality interacts with other OmniSwitch features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

General

- IPv4 and IPv6 packets are sampled on VXLAN Snooping ports. The entire packet header is scanned, but not the payload.

- Fragmented, encrypted, control, or protocol packets (for example, ICMP, LLDP, BPDU) are not supported.

- VXLAN Snooping is applied after other OmniSwitch features, such as Universal Network Profile (UNP), Edge Virtual Bridging (EVB), Learned Port Security (LPS), QoS, DHCP, SPB, and other protocols.

- VXLAN Snooping is a software-based process that uses fast path processing to perform the random packet sampling and QoS policy enforcement. Similar to other features that also require this type of processing, VXLAN Snooping can affect the switch performance and other features can affect the performance of VXLAN Snooping.

Application Fingerprinting (AFP)

AFP and VXLAN Snooping are supported on the same switch port.

- The maximum sampling rate for VXLAN Snooping is 1K pps, but the maximum rate for AFP is 50K pps. If AFP needs more than 1K pps to identify a signature, the processing of that signature may be impacted when VXLAN Snooping is enabled on the same port.

- AFP and VXLAN Snooping also share QoS system resources, which may reduce the amount of such resources that are available for both features.

- If AFP and VXLAN Snooping are both enabled, the allocation of system resources for AFP policies takes precedence. If the QoS policy rules are the same for both features and the actions for these rules do not conflict, then both rules are applied. However, if there is a conflict with policy actions, then the AFP policy rules take precedence.

- AFP is the only other sFlow-based application allowed on a VXLAN Snooping port.

QoS

- VXLAN Snooping uses the QoS infrastructure to create and apply policies to VXLAN packet flows. The following methods are available for applying QoS to VXLAN packet flows:

  – Standard QoS policy rules are configurable for VXLAN Snooping ports and/or VXLAN traffic. These types of rules apply to the outer header information of a VXLAN encapsulated packet.

  – Standard QoS policy rules with a VXLAN policy condition. This type of condition contains the **vxlan** keyword and applies to the inner header information of an encapsulated Ethernet frame in a VXLAN packet.

  – QoS policy lists applied through vNP profiles. When a device is classified into a vNP profile and that profile is associated with a list of QoS policy rules, those rules are applied to the device traffic.

- VXLAN Snooping shares QoS system resources with other OmniSwitch applications. As a result, VXLAN Snooping functionality is subject to the availability of such resources, especially when Application Fingerprinting is also running on VXLAN Snooping ports.

- VXLAN Snooping policies take precedence over other QoS policies if there is a conflict.

- AFP policies apply to the outer headers of VXLAN packets but take precedence over VXLAN Snooping policies in regards to resource allocation on the switch.

sFLOW

VXLAN Snooping and the sFLOW feature are not supported on the same port.

Configuring VXLAN Snooping

This section provides the following information about how to configure and activate the OmniSwitch implementation of VXLAN Snooping:

Configuration Guidelines

Review the guidelines in this section before attempting to configure and activate OmniSwitch VXLAN Snooping.

- Globally enable VXLAN Snooping for the switch before attempting any other VXLAN Snooping command. When enabled, the switch reserves hardware resources for this feature. When disabled, switch resources are released for other purposes.

- VXLAN Snooping is also enabled or disabled at the port level. When this feature is globally enabled for the switch, packet sampling is trigged only on ports that have VXLAN Snooping enabled.

- VXLAN Snooping is intended for use on any switch in a network where VXLAN is used. On these switches, enable VXLAN Snooping on ports that are known to pass VXLAN encapsulated packets.

- VXLAN Snooping shares switch resources with other applications, such as QoS user policies, VLAN Stacking, Application Fingerprinting, DHCP Snooping, Open Flow, IP multicast, and FIP Snooping. Depending on which application comes first, the required hardware resources for other applications may not be available.

- To determine the availability of switch hardware resources and the amount of resources used by VXLAN Snooping policies, use the **show vm-snooping filtering-resource** command. For example:

```
-> show vm-snooping filtering-resource
Total Filtering Resources       :256,
Chassis/Slot    Filtering Resources Used
--------------+---------------------------
  1/SLOT-1         0
  2/SLOT-1         0
```

- When QoS policies with VXLAN conditions are configured on the switch, the policy rules are not programmed into the hardware until a VXLAN packet flow matches the rule. However, if a VXLAN Snooping static policy rule is configured, the rule is programmed into the hardware at the time the rule is created.

- The order in which VXLAN policy rules are programmed into the hardware is based on the order in which the rules were configured. As a result, hardware resources are allocated on a first come, first served basis. To change this, configure VXLAN policy rules with a precedence value. Rules with a higher precedence are allocated resources first, regardless of what order in which the rules were configured.

- To use a QoS policy rule to classify a VXLAN encapsulated packet based on information specific to the packet contents, make sure the rule contains a **vxlan** condition. For example:

```
-> policy condition c1 vxlan vni 1234 udp port 4789
-> policy condition c1 vxlan inner source mac 00:11:22:33:44:00
-> policy condition c1 vxlan inner source ip 10.10.10.10
-> policy action a1 dscp 45
-> policy rule r1 condition c1 action a1 no default-list
```

- Use the **policy condition vxlan** command to configure a specific policy condition for VXLAN packets. The following condition parameters are supported with this command:

  vni
  inner source mac
  inner source mac-group
  inner source ip
  inner source ipv6
  inner ip-protocol
  inner l4-port
  inner vxlan-port

- The **vni** condition parameter is required when configuring a VXLAN Snooping policy condition. The VXLAN header contains the VXLAN Network Identifier (VNI) that is associated with the source MAC address of the Ethernet frame that is encapsulated in a VXLAN packet. The VNI represents the VXLAN segment ID to which the packet belongs.

Enabling/Disabling VXLAN Snooping

By default, the VXLAN Snooping feature is globally disabled for the switch. To enable this feature, use the **vm-snooping admin-state** command with the **enable** option. For example:

```
-> vm-snooping admin-state enable
```

When globally enabled, the VXLAN Snooping process is triggered only on VXLAN Snooping ports and link aggregates. To disable the VXLAN Snooping functionality, use the **vm-snooping admin-state** command with the **disable** option. For example:

```
-> vm-snooping admin-state disable
```

IP packets are not sampled on VXLAN Snooping ports if the feature is globally disabled for the switch.

Changing the VXLAN Snooping Policy Mode

The VXLAN Snooping policy lookup mode determines the allocation of switch resources for VXLAN Snooping policy conditions. There are two types of modes: basic (the default) and advanced. If it is necessary to change the policy mode for the switch, globally disable VXLAN Snooping before making the change.

To change the VXLAN Snooping policy mode setting, use the **vm-snooping policy-mode** command. For example:

```
-> vm-snooping policy-mode advance policy-resource extended inner-header
untagged
```

After making the change, globally enable the VXLAN Snooping feature for the switch. For example, the following commands disable VXLAN Snooping, change the policy mode, and then enable VXLAN Snooping:

```
-> vm-snooping admin-state disable
-> vm-snooping policy-mode advance policy-resource extended
-> vm-snooping admin-state enable
```

The following parameters are used with the **vm-snooping policy-mode** command to tailor the switch resources that are reserved for VXLAN Snooping policy conditions:

Command Parameters	Description
basic	Reserves resources for VNI, VXLAN UDP port, inner source MAC address, and inner IPv4 address policy conditions.
advance	For IPv6 packets, reserves resources for VNI, VXLAN UDP port, inner source MAC address, inner IPv4 source address, IP protocol, and Layer 4 source and destination port policy conditions.
	For IPv6 packets, reserves resources for VNI, VXLAN UDP port, inner source IPv6 address, Layer 4 source and destination port for policy conditions.
inner-header {tagged | untagged | default}	Specifies whether the header of the inner Ethernet frame of the encapsulated packet is tagged or untagged.
policy-resource {extended | default}	Doubles the amount of switch resources, if available, that are reserved for VXLAN Snooping policies.

For example, the following command reserves room for approximately 1024 policy entries that match only inner tagged packets:

```
-> vm-snooping policy-mode basic policy-resource extended inner-header tagged
```

The following command example reserves room for approximately 256 policy entries that match inner tagged packets and 256 policy entries that match inner untagged packets:

```
-> vm-snooping policy-mode basic policy-resource default inner-header default
```

Note. When the VXLAN Snooping policy mode is changed, the QoS entries and the VXLAN Snooping database are flushed.

To determine which VXLAN Snooping policy mode is active for the switch, use the **show vm-snooping config** command.

Configuring Static VXLAN Snooping Policies

VXLAN Snooping policies are not programmed into hardware until a VXLAN packet flow triggers the application of the policy. The advantage to this is that switch resources reserved for VXLAN Snooping are not used until needed. This conserves the usage of switch resources that are allocated for this feature.

To change the default behavior for how switch resources are used for VXLAN Snooping policies, use the **vm-snooping static-policy rule** command. For example, the following command allocates resources for the "r1" QoS policy rule:

```
-> vm-snooping static-policy rule r1
```

In this example, "r1" contains a VXLAN policy condition. If this rule was not configured as a static policy rule, the switch would wait until the rule is applied to a VXLAN packet flow before allocating switch resources reserved for VXLAN Snooping. However, in this case, switch resources are allocated at the time the rule is created, even if there is no packet flow to which the rule is applied.

If the specified rule belongs to a QoS policy list other than the default policy list, then the list name is required when creating the rule. For example:

```
-> vm-snooping static-policy rule r2 list l2
```

In this example, because "r1" is assigned to the "l2" list, the list name is specified with the command. If "r1" was assigned to the system default QoS policy list, then the list name is not required.

To remove a VXLAN Snooping static policy, use the **no** form of the **vm-snooping static-policy rule** command. For example:

```
-> no vm-snooping static-policy rule r1
-> no vm-snooping static-policy rule r2 list l2
```

When the static policy rule is removed, the reserved switch resources that were allocated when the rule was created are made available for VXLAN Snooping operations.

For more information about QoS policy lists, see the "Configuring QoS" chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

Enabling/Disabling VXLAN Snooping Trap Generation

SNMP trap generation can occur when VXLAN Snooping detects and learns a VXLAN packet, the usage of system resources has reached a configurable threshold value, or an NI module or port on which VXLAN packet flows are learned goes down.

By default, VXLAN Snooping trap generation is disabled for the switch. To enable or disable this functionality, use the **vm-snooping trap** command. For example:

```
-> vm-snooping trap enable
-> vm-snooping trap disable
```

Configuring the Filtering Resource Threshold

The filtering resource threshold specifies a percentage of switch resources used by VXLAN Snooping that when reached, will generate an SNMP trap. By default, this threshold is set to 80%. To change the threshold value, use the **vm-snooping filtering-resource trap threshold** command. For example:

```
-> vm-snooping filtering-resource trap threshold 50
```

To set the threshold back to the default value, use the **vm-snooping filtering-resource trap threshold** command with the **default** parameter. For example:

```
-> vm-snooping filtering-resource trap threshold default
```

Configuring the Sampling Rate

The sampling rate determines the rate at which packets received on VXLAN Snooping ports are sampled to determine if they are VXLAN encapsulated packets. This rate is set to 1000 packets-per-second (pps) by default. To change this setting, use the **vm-snooping sampling-rate** command. For example:

```
-> vm-snooping sampling-rate 500
```

The specified sampling rate value is applied to all ports configured as VXLAN Snooping ports.

Configuring the Aging Time

The VXLAN Snooping database contains a list of all the VXLAN packet flows that were sampled. An aging time is associated with the database entries. When the aging time is reached, the flow entry is removed from the database.

By default the VXLAN Snooping aging time is set to 300 seconds. To change the timer value, use the **vm-snooping aging-timer** command. For example:

```
-> vm-snooping aging-timer 100
```

To prevent the database entries from aging out, set the time to zero. For example,

```
-> vm-snooping aging-timer 0
```

Configuring Additional UDP Destination Ports

VXLAN Snooping samples and inspects packets received on VXLAN Snooping ports to discover and learn VXLAN encapsulated packet flows. This function identifies VXLAN packets by checking to see if there is a UDP header in the packet with the destination port set to 4789 (the default).

It is possible to configure VXLAN Snooping to look for additional UDP destination port numbers using the **vm-snooping vxlan udp-port** command. For example:

```
-> vm-snooping vxlan udp-port 8472
```

When a UDP port is added, it does not replace the default 4789 port number. In this example where port 8472 was added, the VXLAN Snooping process will look for both 4789 and 8472 port numbers during the inspection process.

To remove a UDP port number from the list, use the **no** form of the **vm-snooping vxlan udp-port** command. For example:

```
-> no vm-snooping vxlan udp-port 8472
```

Consider the following when configuring additional UDP port numbers:

- The default port number (4789) is not configurable, so it cannot be removed. Only UDP port numbers added through the **vm-snooping vxlan udp-port** command can be removed.

- Avoid using the well-known UDP ports that are already reserved by IANA for other applications.

- Including the default UDP port number (4789), up to eight UDP ports are allowed. However, configuring multiple UDP ports may slow down the VXLAN Snooping process.

- Changing the UDP port number on the fly might stop the VXLAN traffic until the VXLAN Tunnel End Points (VTEPs) in the network are configured with the same destination UDP port.

Configuring VXLAN Snooping Ports

Globally enabling VXLAN Snooping for the switch triggers the sampling of IP packets on VXLAN Snooping ports. The **vm-snooping port** command is used to designate a switch port or link aggregate as a VXLAN Snooping port. For example:

```
-> vm-snooping port 2/1/10
-> vm-snooping linkagg 5
```

To administratively disable packet sampling on the port, use the **vm-snooping port** command with the **admin-state disable** option. For example:

```
-> vm-snooping port 2/1/10 admin-state disable
-> vm-snooping linkagg 5 admin-state disable
```

In this example, packet sampling is stopped but the port and link aggregate are still configured as a VXLAN Snooping port.

Use the **no** form of the **vm-snooping port** command to revert the VXLAN Snooping port back to a regular switch port or link aggregate. For example:

```
-> no vm-snooping port 2/1/10
-> no vm-snooping linkagg 5
```

If VXLAN Snooping is globally disabled for the switch, then none of the VXLAN Snooping ports will sample packets even if the snooping process is administratively enabled on a VXLAN Snooping port or link aggregate.

Configuring Database Entry Logging

The **vm-snooping logging-threshold** command is used to set the number of entries for VXLAN packet flows and hardware statistics that are logged to the **vm_snoop_db_flow_rec.csv** file and the **vm_snoop_hw_stats_rec.csv** file in the **/flash/switch/bridge/vm_snoop/** directory on the local switch. For example, the following command sets the logging threshold to 1000 entries:

```
-> vm-snooping logging-threshold number-of-flows 1000
```

When the number of records logged to the .csv files exceeds the logging threshold value, the corresponding files are renamed to **/flash/switch/bridge/vm_snoop/vm_snoop_db_flow_old_rec.csv** and to **/flash/switch/bridge/vm_snoop_hw_stats_old_rec.csv**.

To set this value back to the default (5000 entries), use the **vm-snooping logging-threshold** command with the **default** option. For example:

```
-> vm-snooping logging-threshold number-of-flows default
```

To turn off logging database entries to the .csv files, set the threshold value to zero. For example,

```
-> vm-snooping logging-threshold number-of-flows 0
```

Note that turning off the logging function does not stop the VXLAN Snooping process from learning VXLAN packet flows and entering the flow information into a database. The **.csv** files are used to maintain a packet flow history and are accessed by Alcatel-Lucent Enterprise network management tools to provide management and visibility for the overlay network traffic.

VXLAN Snooping Configuration Example

The following diagram shows an example of an OmniSwitch VXLAN Snooping implementation that is used to define and apply tenant-based QoS (a different Class of Service for the traffic from each tenant network (Bronze, Silver, and Gold):
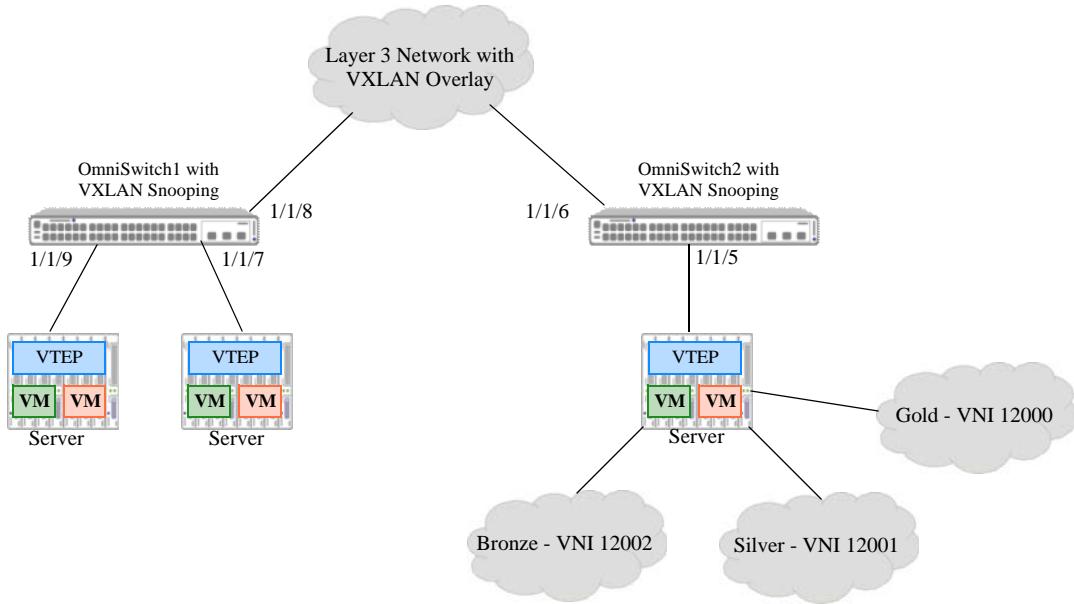
In this example, a Server VTEP (VXLAN Gateway) segments traffic into the following VXLAN Network IDs (VNIs):

- Traffic from the Bronze network is assigned to VNI 12002.

- Traffic from the Silver network is assigned to VNI 12001.

- Traffic from the Gold network is assigned to VNI 12000.

The Server VTEP encapsulates frames from these networks to associate them with their respective VNIs and tunnel the encapsulated packets over the network to other designated VTEPs. VXLAN Snooping is running on OmniSwitch1 and OmniSwitch2 where the VXLAN packets are sampled and QoS is applied through Universal Network Profile (UNP) policy lists.

The following CLI configuration provides a sample of the commands used to configure the VXLAN Snooping functionality on OmniSwitch1 and OmniSwitch2. For more information about how to configure an OmniSwitch as a VTEP, see Chapter 4, "Configuring a VXLAN Gateway."

1 Configure the QoS policy and lists that will apply the tenant-based QoS:

```
-> policy condition BRONZE_12002 vxlan vni 12002
-> policy condition GOLD_12000 vxlan vni 12000
-> policy condition SILVER_12001 vxlan vni 12001
-> policy condition default_any source ip any

-> policy action BRN maximum bandwidth 10K
-> policy action GLD maximum bandwidth 10.0G
-> policy action SIL maximum bandwidth 10.0M
-> policy action allow

-> policy rule GOLD_12000_rule condition GOLD_12000 action GLD no default-list
-> policy rule SILVER_12001_rule condition SILVER_12001 action SIL no default-
list
-> policy rule BRONZE_12002_rule condition BRONZE_12002 action BRN no default-
list
-> policy rule default_allow condition default_any action allow no default-list

-> policy list list1 type unp
-> policy list list1 rules BRONZE_12002_rule GOLD_12000_rule SILVER_12001_rule
default_allow

-> qos apply
```

2 Create a UNP and associate the profile with the QoS policy list created in Step 1:

```
-> unp name tenant-unp vlan 500 qos_policy-list list1
```

3 Enable the UNP functionality on the OmniSwitch1 and OmniSwitch2 ports that will sample VXLAN packets:

```
-> unp port 1/1/9 enable
-> unp port 1/1/7 enable
-> unp port 1/1/8 enable

-> unp port 1/1/5 enable
-> unp port 1/1/6 enable
```

4 Create a UNP classification rule that will classify the VXLAN packets into the UNP created in Step 2:

```
-> unp classification mac-address-range 00:da:95:09:a1:01 00:da:95:09:a5:05 unp-
name tenant-unp
```

5 Globally enable the VXLAN Snooping function for the switch:

```
-> vm-snooping admin-state enable
```

6 Enable VXLAN Snooping on OmniSwitch1 and OmniSwitch2 ports that will sample packets:

```
-> vm-snooping port 1/1/9 admin-state enable
-> vm-snooping port 1/1/7 admin-state enable
-> vm-snooping port 1/1/8 admin-state enable

-> vm-snooping port 1/1/5 admin-state enable
-> vm-snooping port 1/1/6 admin-state enable
```

Verifying the VXLAN Snooping Configuration

A summary of the **show** commands used for verifying the VXLAN Snooping configuration is given here. For some examples of these commands, see "Quick Steps for Configuring VXLAN Snooping" on page 5-4 and the "VXLAN Snooping Configuration Example" on page 5-19.

show vm-snooping config	Displays the global VXLAN Snooping configuration and status for the switch.
show vm-snooping port	Displays the VXLAN Snooping port configuration for the switch.
show vm-snooping database	The contents of the VXLAN Snooping database. Entries include information about the learned VXLAN packet flows. To clear entries from the database, use the **clear vm-snooping database** command.
show vm-snooping virtual-machines	Displays information about devices, such as Virtual Machines, discovered during the VXLAN Snooping process.
show vm-snooping filtering-resource	Displays the amount of switch resources available for and used by VXLAN policy rules.
show vm-snooping statistics	Displays statistics for each VXLAN packet flow on a VXLAN Snooping port or link aggregate. To reset the statistics counters to zero, use the **clear vm-snooping statistics** command.
show vm-snooping static-policy	Displays the static QoS policy rule configuration for VXLAN Snooping.

6 Configuring FIP Snooping

The OmniSwitch implementation of Fibre Channel over Ethernet (FCoE) Initiation Protocol (FIP) snooping supports the FCoE technology used to tunnel Fibre Channel frames encapsulated within Ethernet MAC frames.When FIP snooping functionality is configured and enabled, the OmniSwitch serves as an FCoE transit switch. In this role, the OmniSwitch implementation of Data Center Bridging (DCB) is also used to provide the lossless Ethernet network required to support FCoE.

This implementation of FIP Snooping ensures the security of an FCoE network and maintains a virtual point-to-point network connection between FCoE Nodes (ENodes) and FCoE Forwarder (FCF) devices. An FCoE OmniSwitch is placed between ENodes (FCoE-capable servers) and an FCF to extend the reach of the FCoE network without extending the physical Fibre Channel (FC) connections.

In This Chapter

This chapter describes FCoE and FIP in general and how FIP snooping VLAN and port configurations are applied to the switch. It provides information about configuring FIP snooping through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following topics and procedures are included in this chapter:

- "FIP Snooping Defaults" on page 6-2.

- "Terms and Definitions" on page 6-3.

- "FIP Snooping Overview" on page 6-4.

- "Interaction with Other Features" on page 6-11.

- "Configuring FIP Snooping" on page 6-14.

- "Configuration Example" on page 6-22.

- "Verifying the FIP Snooping Configuration" on page 6-24.

FIP Snooping Defaults

By default, FIP snooping functionality is disabled on the switch. When FIP snooping is enabled, the following global default values are applied:

Parameter Description	Command	Default
FCoE Initialization Protocol (FIP) addressing mode.	**fcoe address-mode**	FPMA (fabric-provided MAC address)
The 802.1p priority value assigned to FCoE traffic.	**fcoe priority**	3
Whether or not only FCoE traffic marked with the FCoE priority value is allowed.	**fcoe priority-protection**	Disabled
Whether or not FCoE traffic marked with a different priority is dropped or remarked (only applied when priority protection is enabled).	**fcoe priority-protection action**	Dropped
Trap threshold value to indicate that a percentage of filtering resources has been used.	**fcoe filtering-resource trap-threshold**	80%
The amount of time the switch waits for keep alive messages from ENodes and FCFs before clearing a FIP session.	**fcoe house-keeping-time-period**	300 seconds

Terms and Definitions

The OmniSwitch implementation of FIP snooping and DCB allows the switch to operate as an FCoE transit switch in an FCoE network. The following terms and definitions provided in this section describe FCoE network components discussed in this chapter.

- **CNA (Converged Network Adapter)**: A network interface card (NIC) that combines the functionality of a FC host bus adapter (HBA) and a lossless Ethernet network interface card (NIC). A CNA is used to connect servers to a FC storage area network (SAN) and an Ethernet network.

- **DCB (Data Center Bridging)**: A set of standards that define protocols to extend the capabilities of Ethernet to support the convergence of data and storage in the data center. DCB is required to provide the lossless Ethernet transport required to support FCoE (see Chapter 2, "Configuring Data Center Bridging," for more information).

- **ENode (FCoE Node)**: The FCoE functionality of a CNA that instantiates virtual FC interfaces (VN_Ports) to connect to virtual FC interfaces (VF_Ports) on an FCF switch. The virtual links established between the ENode and FCF allows FCoE servers attached to an Ethernet network to access storage devices in the SAN.

- **Fibre Channel (FC)**: A high-speed network technology primarily used for storage networking in the data center. FC is used to pool disk storage for all hosts in the network, which helps to provide a centralized storage solution that is easier to manage and more reliable.

- **FCoE (Fibre Channel over Ethernet)**: A standard (T11 FC-BB-5) that defines a virtual FC link topology and encapsulation for transporting native FC frames over an Ethernet network. FCoE frames are identified by Ethertype 0x8906 to differentiate them from other traffic on the same network.

- **FCF (FCoE Forwarder)**: A switch capable of transmitting native FC frames and FCoE frames. The FCF sits at the edge of a FC SAN to translate and forward traffic between an FCoE network and the FC SAN.

- **FCoE Initialization Protocol (FIP)**: An Ethernet Layer 2 control protocol that establishes and maintains virtual FC links between pairs of ENodes and FCFs over the FCoE network. FIP control frames do not carry any payload and are identified by Ethertype 0x8914 to differentiate them from FCoE data frames and other Ethernet traffic on the same network.

- **FIP Snooping**: A security mechanism implemented on FCoE transit switches. FIP snooping inspects FIP frames received on transit switch ports and then dynamically builds ACLs based on the data in those frames. The ACLs filter FIP frames received on FCoE ports associated with FCoE VLANs and will only allow traffic from FCoE sources that successfully logged into the FCoE fabric.

- **FCoE Transit Switch**: A Layer 2 DCB switch capable of transporting encapsulated FCoE frames. May also use FIP snooping to secure FCoE traffic.

- **FCoE VLAN**: A type of VLAN that is dedicated to carrying only FCoE and FIP traffic. ENodes use this type of VLAN to transmit and receive FCoE and FIP traffic and establish virtual links to the FCF. Only FCoE ports are tagged with this VLAN and all other non-FCoE traffic should be segregated into other VLAN types.

- **VF_Port (Virtual Fabric Port)**: The virtual FCoE counterpart to a F_Port in a native FC network. An FCF switch instantiates VF_Ports to connect with ENode VN_Ports to establish a virtual point-to-point link between the ENode and FCF.

- **VN_Port (Virtual Node Port)**: The virtual FCoE counterpart to an N_port in a native FC network. ENodes instantiate VN_Ports to connect with FCF VF_Ports to establish a virtual point-to-point link between the ENode and FCF.

FIP Snooping Overview

FCoE transit switches are used to extend the reach of an FCoE network without having to use additional, more expensive Fibre Channel (FC) equipment that is usually required to extend the native FC network. The OmniSwitch can operate as an FCoE transit switch within a multi-hop FCoE network.

To provide the necessary FCoE transit switch functionality, the OmniSwitch supports FCoE Initialization Protocol (FIP) snooping and Data Center Bridging (DCB) protocols. Both FIP snooping and DCB are required to provide a secure, lossless Ethernet fabric for FCoE traffic.A transit switch is basically a Layer 2 DCB switch that bridges encapsulated FCoE traffic over the Ethernet fabric between FCoE end devices.

FCoE is a standard (T11 FC-BB-5) that builds a topology of virtual FC links on top of a physical Ethernet network. To transport native FC traffic across this topology, FCoE encapsulates native FC frames into Ethernet MAC frames, which are then forwarded through the Ethernet network in the same manner as any other Ethernet frame. As a result, FCoE supports the convergence of FC and Ethernet traffic on the same physical Ethernet network.

Encapsulated FC frames are forwarded along the Ethernet transport between two types of FCoE endpoints: an FCoE Node (ENode) and an FCoE Forwarder (FCF). An ENode refers to the FCoE capability of a converged network adapter (CNA). An FCF is a switch that is capable of transmitting FCoE frames onto the FCoE network and native FC frames onto the FC storage area network (SAN). Both ENodes and FCFs handle the encapsulation and de-encapsulation of FC frames.

It is important to remember that an FCoE transit switch forwards native FC frames that are already encapsulated into Ethernet frames. The transit switch does *not* encapsulate or de-encapsulate native FC frames.

- The ENode and FCF devices in the FCoE fabric are responsible for encapsulating native FC frames into Ethernet and then forwarding those frames onto the FCoE transit switch path.

- FCoE encapsulated frames received by an ENode or FCF are then de-encapsulated and the native FC frames are forwarded onto the intended FC network destination.

The encapsulation of native FC frames into Ethernet allows the use of transit switches between ENodes and FCFs to extend the reach of the FCoE network. An Ethernet switch can participate in an FCoE network as an FCoE Forwarder switch or as an FCoE transit switch.

- An FCoE Forwarder switch encapsulates native FC frames into Ethernet frames and then forwards those frames over the FCoE network. When the FCF receives encapsulated frames, the FCF strips the Ethernet encapsulation and then forwards the native FC frame on the FC network.

- An FCoE transit switch is an access switch that transports encapsulated FCoE frames between ENodes and FCFs. A transit switch does not encapsulate or de-encapsulate native FC frames; the switch acts as a pass-through switch in an FCoE topology.

Note. The OmniSwitch can participate in an FCoE network as an FCoE transit switch or as an FCoE Forwarder. See Chapter 7, "Configuring an FCoE/FC Gateway," for more information about using the OmniSwitch as an FCoE Forwarder.

As an FCoE transit switch, the OmniSwitch will sit in between ENodes and an FCF or OmniSwitch FCoE/FC gateway as part of a multi-hop FCoE network, as shown in Figure 1 on page 6-5. In this role, the OmniSwitch is a pass-through switch that is transparent to the ENode and FCF endpoints.
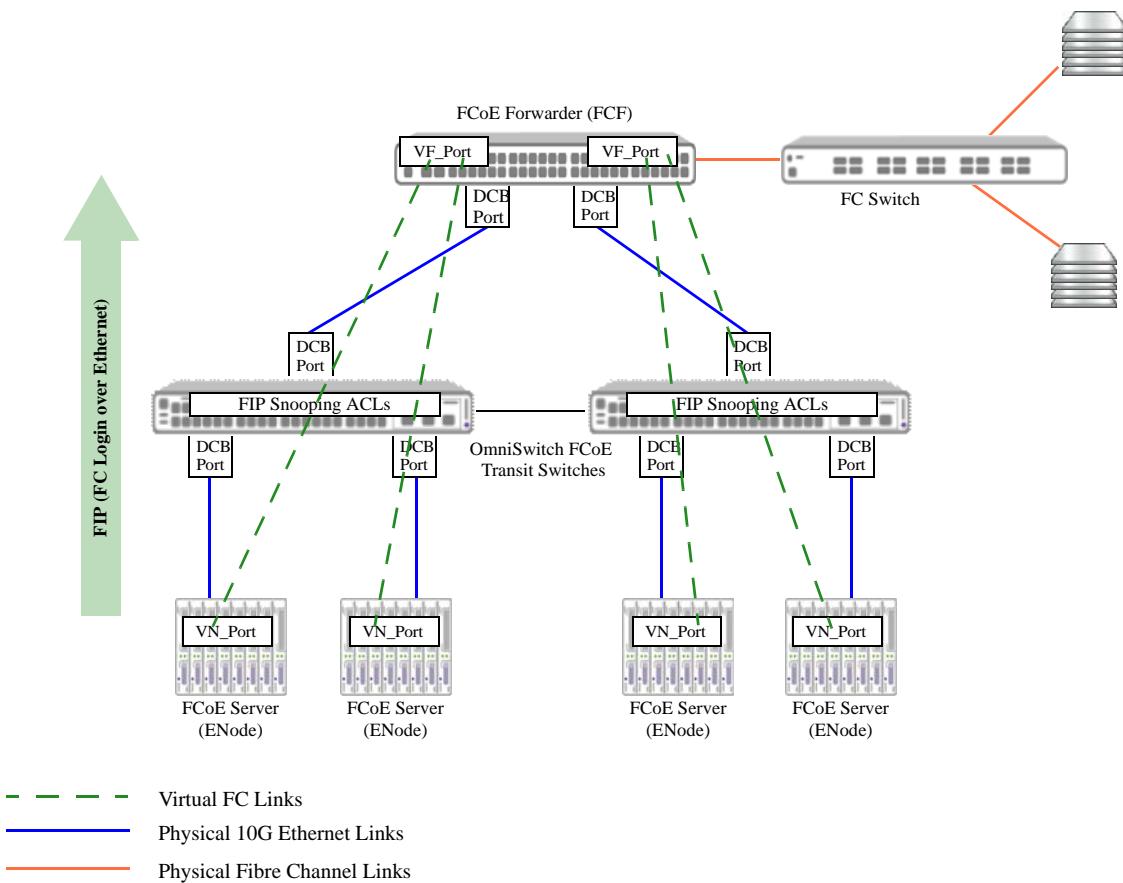
Figure 1: OmniSwitch Transit Switch Connecting FCoE Endpoints

The virtual FC links between ENode VN_Ports and FCF VF_Ports are discovered, created, and maintained by the FCoE Initialization Protocol (FIP). Based on FIP requests and responses traversing these links, the transit OmniSwitch will dynamically create ACLs to filter the FIP traffic received on the switch FCoE ports connected to FCoE devices. These ACLs perform the FIP snooping function that protects the FCoE network from unauthorized access and data transmission through the transit switch.

FCoE Initialization Protocol

Extending the reach of an FC network using FCoE requires FCoE to provide the same point-to-point connectivity and security that is required within a native FC network. While FCoE encapsulation handles the forwarding of native FC frames over 10G (or higher) Ethernet, the FCoE Initialization Protocol (FIP) is needed to provide the point-to-point connectivity and base FC device login process over Ethernet.

FIP is a control protocol that establishes and maintains virtual FC links between ENode VN_Ports and FCF VF_Ports in the FCoE network. These virtual links can traverse a multi-hop FCoE transit switch path but appear as direct point-to-point connections to the FCF. The underlying FCoE Ethernet link path is invisible to the FCF. FIP allows ENodes the ability to discover FCFs and complete the FC login process over the FC virtual links.

The remainder of this section describes the major functions FIP performs to support the FC login process over FCoE.

FIP VLAN Discovery

ENodes send out FIP VLAN discovery requests to find FCoE VLANs on which the ENode can transmit and receive other FIP protocol traffic and FCoE encapsulated frames.

- The ENode sends FIP VLAN discovery requests untagged to the ALL-FCF-MACs multicast address. Because the request packets are sent untagged, the VLAN discovery process takes place on the default VLAN. On the OmniSwitch, this is the default VLAN assigned to the FCoE ports.

- FCFs also listen to the ALL-FCF-MACs multicast address. Those FCFs associated with the same default VLAN as the ENode will respond to the request with a list of available FCoE VLANs for ENode VN_Port login.

- The ENode selects one of the FCoE VLANs received in the FCF notifications and then continues the FIP process on that VLAN.

FIP discovery requests are the only FIP packets carried on the default VLAN. All other FIP functions and FCoE run on the discovered FCoE VLANs.

FIP FCF Discovery

Once an ENode has selected an FCoE VLAN, the next step is to discover FCFs that belong to the same VLAN and then select an FCF that is available for logins. There are two ways FCF discovery can occur:

- An FCF initiates discovery by periodically sending multicast FIP FCF discovery advertisements on each configured FCoE VLAN. These advertisements are sent to the ALL-ENODE-MACs and are used by the FCFs to notify ENodes that the FCF has VF_Ports available for establishing virtual links with ENode VN_Ports.

- When a new ENode comes online, the ENode sends a unicast FIP FCF discovery solicitation message on the FCoE VLAN to which the ENode belongs. The solicitation message is sent to the ALL-FCF-MACs multicast address to request a unicast advertisement. An FCF that receives the solicitation can send a unicast FIP FCF discovery advertisement back to the ENode that initiated the request.

Once an ENode has gathered all of the FIP FCF discovery advertisements, the ENode selects an FCF to contact for fabric login and virtual link establishment.

FIP Login

When an ENode selects an FCF, the ENode sends a fabric login (FLOGI) request to the FCF. The FCF acknowledges the request and, depending on the MAC addressing mode used, the FCF sends the ENode a locally unique MAC address to use for FCoE encapsulation.

There are two MAC addressing modes that ENodes and FCFs can use to provide a locally unique MAC address for FCoE operations: fabric-provided MAC address (FPMA) mode or server-provided MAC address (SPMA) mode. The OmniSwitch supports both FPMA and SPMA modes.

- When the FPMA mode is in use by the ENode, the MAC address assigned to the ENode VN_Port is a 48-bit address, which consists of the 24-bit FCoE mapped address prefix (FC-MAP) combined with a 24-bit FC identifier (FCID). The FC-MAP is a configurable value on OmniSwitch FCoE VLANs. In this mode, the ENode can have different VN_Ports, each with their own unique MAC address.

- When the SPMA mode is in use by the ENode, the server provides the MAC address to the FCF, which then determines if the SPMA is an address approved for FCoE access.

The ENode and the FCF must use the same addressing mode to establish virtual links between the ENode and FCF. Note that the ENode continues to use the globally unique ENode MAC address (CNA MAC) for FIP control operations; the FPMA or SPMA is used for FCoE frame transactions.

A successful ENode fabric login establishes a new virtual link between the ENode VN_Port and FCF VF_Port. FCoE servers attached to an Ethernet network can access storage devices in the FC SAN by exchanging FCoE frames (encapsulated FC payloads) over the virtual link.

OmniSwitch FIP Snooping ACLs

FIP snooping is a security mechanism used in multi-hop FCoE networks to ensure that only traffic from ENodes that have successfully logged into the FC fabric passes through the FCoE ports and VLANs on the transit switch and reaches the FCF. In other words, securing the dedicated FC virtual point-to-point link that is established between an ENode and FCF through the FIP login process to prevent unauthorized access to the FC network.

Based on Annex C (*Increasing FC-BB_E Robustness Using Access Control Lists*) of the T11 FC-BB-5 standard, the OmniSwitch implementation of FIP Snooping filters FIP traffic on FCoE ports using QoS ACLs. The OmniSwitch QoS feature inspects FIP packets and dynamically configures FIP snooping ACL entries based on the contents of those packets. The ACLs are then applied to determine whether frames are forwarded or discarded on OmniSwitch FCoE ports.

- The FCoE ACLs take precedence over other QoS policies.

- If a frame matches multiple FCoE ACL entries programmed in the hardware, the first matching ACL entry in the hardware is applied to the frame.

When FIP snooping is globally enabled for the transit OmniSwitch, the switch will deny traffic from any ENode on any FCoE VLAN until the ENode successfully logs on to the FC fabric.

FCoE Port Roles

The dynamic ACL configuration applied to an OmniSwitch FCoE port is determined by the role the port is assigned within the FCoE network. A port is assigned one of the following roles when the port is configured as an FCoE port:

- **Edge port**—a direct link to an ENode host (ENode-facing port).

- **ENode-only port**—a link between FCoE transit switches that carries traffic from ENode to FCF but not the reverse.

- **FCF-only port**—a link between FCoE transit switches that carries traffic from FCF to ENode but not the reverse.

- **Mixed port**—a link between FCoE transit switches that carries traffic in both directions (from FCF to ENode or from ENode to FCF).

- **Trusted port**—the FCoE port is trusted; traffic on this port is not filtered by FIP ACLs. This port role type is typically assigned to the switch FCoE port that connects to an Ethernet port on the FCF.

The port role assigned depends on where the port resides within the FCoE transit switch path and the source of the traffic (ENode, FCF, or both) the port will receive. Although FCoE ports can pass bidirectional FCoE traffic, the dynamic ACL configuration is based on the ingress traffic.

The remaining subsections provide an example of port role locations within the network and the ACL functionality applied for the given port role.

Edge Port

The following diagram shows an example of where an FCoE edge port is placed in the FCoE network topology:
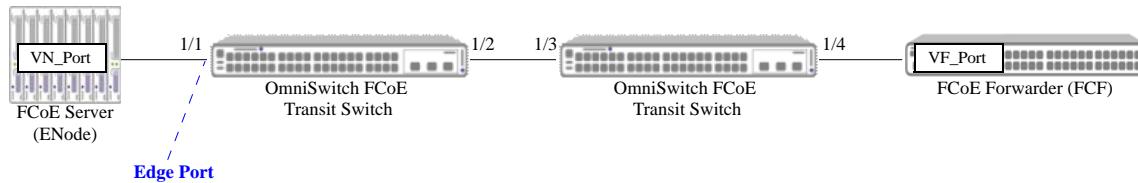
Figure 2: FCoE Port Role–Edge

ACL entries are applied on an ENode-facing edge port to:

- Enable transmission of FCoE frames from ENodes to FCF.

- Ensure that only FIP frames are sent to the FCF.

- Prevent transmission of FCoE frames from ENodes prior to fabric login (FLOGI).

- Prevent transmission of FCoE frames from one ENode to another ENode.

- Ensure that after fabric login (FLOGI) the ENode only uses FCoE source MAC addresses assigned by the FCF.

- Ensure that after FLOGI the assigned source MAC address is only used for FCoE traffic.

- Ensure that after the FLOGI that FCoE frames are only addressed to the FCF.

- Ensure that ACL entries do not interfere with other switch ACLs.

- Ensure that the ACL entries do not restrict non-FCoE traffic (traffic that does not match the FCoE Ethertype and does not use an FCoE source MAC address).

ENode-only Port

The following diagram shows an example of where an FCoE ENode-only port (port that carries ingress traffic from the ENode) is placed in the FCoE network topology:
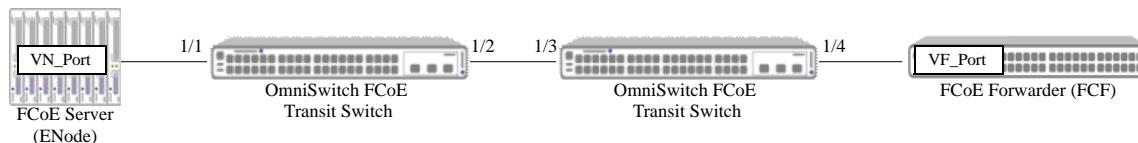
Figure 3: FCoE Port Role–ENode

ACL entries are applied on an ENode-only port to:

- Verify that all FIP frames are addressed to the FCF.

- Verify that all FCoE frames are addressed to the FCF.

- Verify that all FCoE frames are only sourced by ENodes (if FPMA is in use).

- Prevent FCoE traffic from flowing between FCF (optional).

FCF-only Port

The following diagram shows an example of where an FCoE FCF-only port (port that carries ingress traffic from the FCF) is placed in the FCoE network topology:
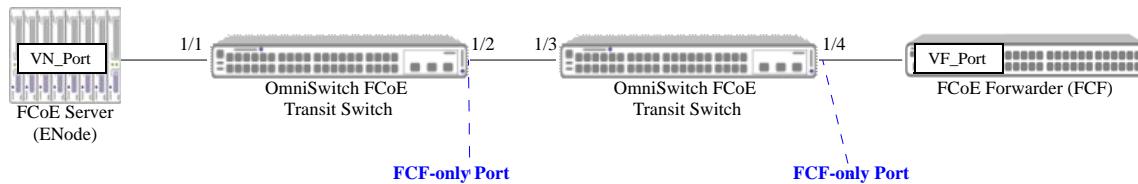
Figure 4: FCoE Port Role–FCF

ACL entries are applied on an FCF-only port to:

- Verify that all FIP frames are sourced from an FCF and are not destined to another FCF.

- Verify that all FCoE frames are sourced from an FCF.

- Verify that all FCoE frames are destined to an ENode (FPMA only).

- Prevent the forwarding of FCoE frames to an FCF (optional).

Mixed Port

The following diagram shows an example of where an FCoE mixed port (port that carries FCF-to-ENode traffic and ENode-to-FCF traffic) is placed in the FCoE network topology:
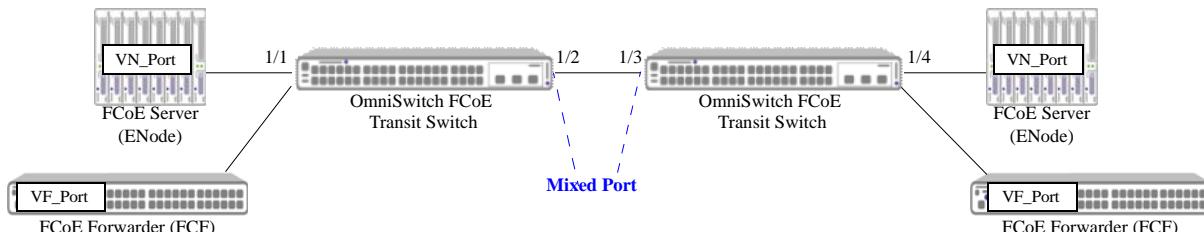
Figure 5: FCoE Port Role–Mixed

ACL entries are applied on mixed port to:

- Verify that all FIP frames are sourced from or destined to an FCF.

- Verify that all FCoE frames are sourced from an FCF and are sent to an ENode or FCF, or verify that all FCoE frames are sourced from an ENode and are sent to an FCF (FPMA only).

- Verify that all FCoE frames are sourced from or sent to an FCF.

Trusted Port

The following diagram shows an example of where an FCoE trusted port is placed in the FCoE network topology:
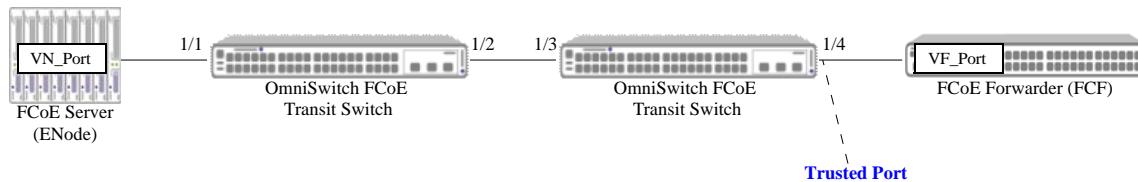
Figure 6: FCoE Port Role–Trusted

No ACLs are applied when the port is configured as an FCoE trusted port.

Interaction with Other Features

This section contains important information about how other OmniSwitch features interact with the FCoE and FIP Snooping features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

Data Center Bridging (DCB)

FCoE requires an underlying lossless Ethernet network. The OmniSwitch supports the following Data Center Bridging (DCB) protocols that extend Ethernet capabilities to support the convergence of storage and data in virtualized networks:

* **Priority-Based Flow Control (PFC)**—Based on the IEEE 802.1Qbb standard, PFC pauses traffic based on congestion priority instead of blocking the entire link when congestion occurs. Allows lossless and lossy traffic with different priorities on the same physical port.

* **Enhanced Transmission Selection (ETS)**—Based on the IEEE 802.1Qaz standard, ETS provides a common framework for dynamic bandwidth management. ETS groups related traffic into priority groups (Traffic Classes) to which bandwidth guarantees and scheduling are applied.

* **Data Center Bridging Exchange (DCBX)**—Based on the IEEE 802.1Qaz standard, DCBX uses the Link Layer Discovery Protocol (LLDP) to exchange and negotiate PFC and ETS information between two directly connected peer switches. Enabled by default, DCBX is responsible for auto-negotiation and auto-configuration of link parameters for DCB functions.

The OmniSwitch implementation of DCB provides enhanced QoS congestion and bandwidth allocation to support multiple traffic types on the same Ethernet link. The supported DCB protocols use the Class of Service (CoS) 802.1p markings found in the frame header to define traffic groups. These markings are the result of QoS classification that occurs prior to and separately from the application of DCB functionality. DCB does not replace existing QoS classification and enqueuing mechanisms.

On the OmniSwitch, embedded port profiles are used to apply the supported PFC, ETS, and DCBx configuration to traffic flows. This approach is similar to how QSet profiles are used to apply the QoS configuration for bandwidth management and egress port queue scheduling. However, DCB and QSet profiles are mutually exclusive on the same port.

By default, when the OmniSwitch Data Center software license is installed, DCB profile 8 (DCP 8) is applied to all switch ports. In addition, DCBx is enabled on each switch port. However, DCP 8 applies lossy priority flow control to ingress traffic flows. However, the FCoE transit switch path requires a lossless Ethernet transport. As a result, it is necessary to apply to each participating FCoE port one of the other pre-defined profiles or create a custom profile that guarantees lossless priority flow control for the 802.1p value that is used to mark FCoE traffic.

For more information about DCB profiles, see Chapter 2, "Configuring Data Center Bridging," of the *OmniSwitch AOS Release 8 Data Center Switching Guide*.

For more information about CoS 802.1p priority bit classification and marking, see "How Traffic is Classified and Marked" on page 26-5 in Chapter 26, "Configuring QoS," of the *OmniSwitch AOS Release 8 Network Configuration Guide*.

802.1AB Link Layer Discovery Protocol

DCBX uses the IEEE 802.1AB Link Layer Discover Protocol (LLDP) to discover and exchange Ethernet capabilities between two link peers (such as between an ENode and an edge FCoE switch). There are specific LLDP Type-Length-Values (TLVs) that DCBx defines for exchanging PFC and ETS attribute values.

In addition to the DCBX TLVs, the OmniSwitch implementation of LLDP also supports the Application Priority TLV, which is used to advertise the designated FCoE priority to a peer device. Activating the Application Priority TLV and specifying the FCoE priority value advertised through this informational TLV is configured using LLDP CLI commands. Make sure that the priority value advertised through the Application Priority TLV matches the priority value designated for FCoE traffic classification and priority flow control.

Loop Avoidance

Configuring MSTP or ERP for loop avoidance is recommended. When using MSTP assign each FCoE VLAN its own MST ID. This will improve performance in a topology where each VSAN assigned to a VLAN uses different FCFs.

Universal Network Profile (UNP)

- Configuring FCoE on a UNP port is allowed. UNP can dynamically assign an FCoE port to a default VLAN, but manually tagging the port with the FCoE VLAN is still required.

- An FCoE VLAN cannot be assigned to a UNP.

- Enabling UNP on an FCoE port that is tagged with non-FCoE VLANs is not allowed. Enabling UNP on an FCoE port is allowed only on FCoE ports that are tagged with FCoE VLANs.

- Changes to a UNP port triggers a MAC flush of all MAC addresses learned on that port, including FIP and FCoE MAC addresses if the UNP port is also an FCoE port.

- When UNP is enabled or disabled on an FCoE port, all MAC addresses are flushed, including FIP and FCoE MAC addresses.

For more information about UNP, see Chapter 28, "Configuring Access Guardian."

QoS

The QoS application is responsible for the dynamic configuration of the ACLs that FIP snooping uses to secure the traffic flowing through an OmniSwitch FCoE transit switch. As a result FIP snooping ACLs and other QoS functions share the same system resources. This means that the number of FIP snooping sessions allowed is subject to the availability of the shared resources.

The following limitations apply when FCoE is configured on a switch port:

- FCoE port configuration overrides global QoS settings.

- If there are no QoS system resources available for FIP snooping, an error is given indicating that resources were not allocated.

- To achieve priority protection of the FCoE traffic, if any of the non-FCoE traffic with the FCoE priority is received on the FCoE port, the traffic is remarked to a priority lower than that of the default

FCoE priority and the configured FCoE priority. Other lossless traffic will have its own priority and should not clash with the FCoE priority.

Multiple VLAN Registration Protocol (MVRP)

- The FCoE transit switch path between ENodes and the FCF requires configuration of the same FCoE VLAN on each switch. To configure FCoE VLANs dynamically through the core network that connects FCF and ENodes, enable MVRP on all FCoE ports. However, on edge switches that are connected to an ENode or FCF, statically configuring the FCoE VLANs and tagging them with the appropriate FCoE ports is still required.

- If an MVRP join message is received on a FCOE VLAN, no VPA is created whether or not the receiving port is an FCoE port. However, as part of an MVRP advertisement, the same VLAN will be transmitted so that it can be learned on core switches.

- Manually configuring the FCoE VLAN end-to-end is recommended, but if there are switches that are not directly connected to ENodes or an FCF and they are secure, then MVRP may help reduce FCoE FIP snooping configuration steps.

Configuring FIP Snooping

Configuring OmniSwitch FIP snooping in an FCoE multi-hop network requires the following general steps:

1 Configure the DCB lossless Ethernet transport. The following steps are required on each transit OmniSwitch to ensure lossless transport of FCoE traffic through the multi-hop FCoE network.

a. Select a pre-defined DCB profile or create a custom profile that defines lossless priority flow control for a specific CoS priority value. By default, FCoE CoS is set to 3.

b. Determine the end-to-end lossless data path between ENodes and the FCF and apply the profile identified in Step 1a to each port that will forward FCoE traffic. For example, the following command assigns DCB profile 11 to ports 8/1 and link aggregate 10.

```
-> qos qsi port 8/1 qsp dcb dcp-11
-> qos qsi linkagg 10 qsp dcb dcp-11
```

In addition, make sure that the DCBx protocol is enabled on each participating port. DCBx is enabled by default on all switch ports, but the following command is used to enable DCBx, if necessary:

```
-> qos qsi port 8/1 dcb dcbx admin-state enable
-> qos qsi linkagg 10 dcb dcbx admin-state enable
```

c. Make sure that the traffic requiring lossless transmission is marked with the same CoS priority value to which the DCB profile will apply lossless priority flow control. The switch will only pause frames sent with the same priority value as the lossless priority value specified in the DCB profile.

d. Make sure that all participating FCoE ports are configured as QoS trusted ports with the default classification set to 802.1p. For example:

```
-> qos port 8/1 trusted default classification 802.1p
-> qos linkagg 10 trusted default classification 802.1p
```

e. Configure 802.1ab LLDP on FCoE ports to send the application-priority TLV. This is done to provide the peer device with the priority value set for FCoE traffic. For example:

```
-> lldp port 8/1 tlv application enable
-> lldp port 8/1 tlv application priority 3
```

2 Configure FCoE VLANs. Create the same FCoE VLAN on each participating OmniSwitch. For example:

```
-> fcoe vlan 100
```

3 Configure FCoE ports. Configure each port that will carry FCoE/FIP traffic as an FCoE port, which includes specifying the FCoE role (Edge, ENode-only, FCF-only, Mixed, Trusted) for the port. For example, port 8/1 connects to an ENode and link aggregate 10 will forward ingress traffic from the FCF.

```
-> fcoe port 8/1 role edge
-> fcoe linkagg 10 role fcf-only
```

4 Assign FCoE ports to FCoE VLANs. Tag each FCoE port with the same FCoE VLAN on each participating OmniSwitch. For example:

```
-> vlan 100 members port 8/1 tagged
-> vlan 100 members linkagg 10 tagged
```

5 Enable FIP snooping. Administratively enable FIP snooping on each participating OmniSwitch. For example:

```
-> fcoe fip-snooping admin-state enable
```

Additional FIP snooping global, VLAN, and port parameters are configurable to fine tune FIP snooping on the OmniSwitch. These parameters have default values that apply when FIP snooping is enabled for the switch. For more information about configuring FIP snooping parameters, see the following sections:

- "Configuration Guidelines" on page 6-15.

- "Configuring Lossless DCB for FCoE" on page 6-16.

- "Configuring Global FCoE Parameters" on page 6-18.

- "Configuring FCoE VLANs" on page 6-20.

- "Configuring FCoE Ports" on page 6-21.

- "Configuration Example" on page 6-22.

For more information about DCB profiles, see Chapter 2, "Configuring Data Center Bridging," of the *OmniSwitch AOS Release 8 Data Center Switching Guide*.

For more information about CoS 802.1p priority bit classification and marking, see "How Traffic is Classified and Marked" on page 26-5 in Chapter 26, "Configuring QoS," of the *OmniSwitch AOS Release 8 Network Configuration Guide*.

Configuration Guidelines

Review the guidelines in this section before attempting to configure an OmniSwitch FIP snooping transit switch path between ENodes and FCFs.

FCoE VLANs

- Manual configuration of the FCoE VLAN is required along the transit switch path and on the FCF. However, the ENode may invoke FIP VLAN discovery to discover the FCoE VLANs within the transit path. If not, manual configuration of the FCoE VLANs may be required on the appropriate ENodes.

- The ENode, transit switches, and the FCF must all use the same addressing mode (FPMA or SPMA) to establish virtual links between the ENode and FCF. On the OmniSwitch, the addressing mode is a configurable parameter for the FCoE VLAN. Make sure each OmniSwitch FCoE VLAN in the transit switch path is configured with the same dress mode used by the ENode and FCF.

- FCoE VLANs must be configured with an MTU size of at least 2500 bytes to accommodate FCoE frames, which are larger than the standard Ethernet frame size. Make sure the MTU size is configured end-to-end to prevent fragmentation of FCoE frames.

- Configuring an FCoE VLAN as a default VLAN for a port or link aggregate is not allowed. In addition, configuring default VLAN 1 as an FCoE VLAN is not allowed.

- The following features are not supported on FCoE VLANs:
    - IGMP Snooping
    - IP interface
    - HA VLANs
    - SVLAN and CVLAN
    - Shortest Path Bridging (SPB)
    - MCLAG
    - UDP Relay, DHCP Snooping
    - Universal Network Profile (UNP)

FCoE Ports

- FCoE is only supported on 10G or faster ports that are associated with an FCoE lossless DCB profile. In addition, DCBX must be enabled on the port with both PFC and ETS in an active state (either forced or negotiated via DCBX). The DCB configuration is done separately using QoS port and profile commands.

- FCoE ports must be manually assigned to a default VLAN and then tagged with the FCoE VLAN that will forward the FCoE and FIP frames on that port.

- The default VLAN for the FCoE ports will carry FIP VLAN discovery requests through FCoE network. As a result, it is necessary to configure the same default VLAN for FCoE ports on each switch. All other FIP traffic and FCoE frames are carried on the FCoE VLAN.

The following features are not supported on FCoE ports:

- Learned Port Security (LPS)

- Port Mirroring and Remote Port Mirroring

- Shortest Path Bridging (SPB)

- Edge Virtual Bridging (EVB).

FCoE Priority

Make sure to use the same FCoE priority value when configuring the following:

- The FCoE priority value used to classify FCoE traffic (QoS class of service markings).

- The lossless priority value defined in the DCB profile applied to participating FCoE ports.

- The FCoE priority value defined in the LLDP Application Priority TLV that is advertised on FCoE edge ports connected to ENodes.

- The FCoE priority that is protected when FCoE priority protection is enabled for the switch. Priority protection is disabled by default.

Configuring Lossless DCB for FCoE

The OmniSwitch Data Center Bridging (DCB) configuration is applied using DCB profiles. By default, DCB profile 8 (DCP 8) is applied to all switch ports and link aggregates. However, DCP 8 does not apply lossless priority flow control; all traffic classes are set to lossy. The FCoE transit switch path requires a

lossless Ethernet transport. As a result, it is necessary to apply a profile that defines lossless flow control for the desired FCoE traffic priority on all of the following participating ports:

- FCoE edge ports that connect to ENodes (servers).

- FCF only ports that connect to FCoE Forwarders (FCFs).

- All port connections between OmniSwitch FCoE transit switches.

There are 11 predefined DCB profiles (DCP 1–11) available, and some of these profiles provide lossless traffic classes for one or more priorities. However, these profiles may not provide lossless traffic classes for the designated FCoE traffic priority or may provide lossless traffic classes for all priorities (DCP 7, 9, and 11). In this case, creating and applying a custom DCB profile, as follows, is recommended:

1 Create the new custom profile based on DCP 9 or 11. For example:

```
-> qos qsp dcb dcp-12 import qsp dcb dcp-11
```

2 Configure all traffic classes in DCP 12 to lossy, except for the priority designated for lossless FCoE traffic. In this example, priority 3 is the FCoE priority so its traffic class is not changed to lossy:

```
-> qos qsp dcb dcp-12 tc 0 pfc flow-type nLL
-> qos qsp dcb dcp-12 tc 1 pfc flow-type nLL
-> qos qsp dcb dcp-12 tc 2 pfc flow-type nLL
-> qos qsp dcb dcp-12 tc 4 pfc flow-type nLL
-> qos qsp dcb dcp-12 tc 5 pfc flow-type nLL
-> qos qsp dcb dcp-12 tc 6 pfc flow-type nLL
-> qos qsp dcb dcp-12 tc 7 pfc flow-type nLL
```

3 Apply custom profile DCP 12 to all participating FCoE ports. For example:

```
-> qos qsi port 1/5/16 qsp dcb dcp-12
-> qos qsi linkagg 5 qsp dcb dcp-12
```

Note. If FCoE priority protection is enabled for the switch, make sure the DCB profile used on participating FCoE ports ensures lossless traffic classes for the protected priority values (see "Enabling/Disabling FCoE Priority Protection" on page 6-18).

For more information about DCB profiles, including predefined profile definitions, see Chapter 2, "Configuring Data Center Bridging," of the *OmniSwitch AOS Release 8 Data Center Switching Guide*.

Configure the LLDP Application Priority TLV

DCBX uses the IEEE 802.1AB Link Layer Discover Protocol (LLDP) to discover and exchange Ethernet capabilities between two link peers (such as between an ENode and an edge FCoE switch). There are specific LLDP Type-Length-Values (TLVs) that DCBx defines for exchanging PFC and ETS attribute values.

In addition to the DCBX TLVs, the OmniSwitch implementation of LLDP also supports the Application Priority TLV. In an FCoE network, this TLV is used to advertise the designated FCoE priority to a server (ENode) directly connected to an FCoE edge port.

Activating the Application Priority TLV and specifying the FCoE priority value to advertise is configured using **lldp tlv application** command with the **fcoe priority** option. For example:

```
-> lldp port 1/5/16 tlv application fcoe priority 3
-> lldp port 2/4/16 tlv application fcoe priority 3
```

Make sure the FCoE priority value advertised with the Application Priority TLV is the same priority value that is designated for FCoE traffic in the network configuration.

Configuring Global FCoE Parameters

This section describes the global FIP snooping configuration, which includes enabling and disabling FIP snooping, changing the FIP addressing mode, configuring the FCoE traffic priority, setting the filtering resource threshold value, and configuring the housekeeping time period.

Enabling/Disabling FIP Snooping

By default FIP snooping is disabled on the switch. To enable FIP snooping, use the **fcoe fip-snooping** command. For example:

```
-> fcoe fip-snooping admin-state enable
```

When FIP snooping is enabled, FCoE traffic is dropped on all switch VLANs and ports that are *not* configured as FCoE VLANs and ports.

To disable FIP snooping, enter the **fcoe fip-snooping** command with **disable** option:

```
-> fcoe fip-snooping admin-state disable
```

Changing the FIP Addressing Mode

The FIP addressing mode determines if the MAC address assigned to an ENode is a fabric-provided MAC address (FPMA) assigned by the FCF or a server-provided MAC address (SPMA) assigned by the ENode server but verified by the FCF. The MAC address is assigned during the FIP login process and is used to identify the ENode for FCoE frame transactions and FC virtual link termination.

Consider the following when changing the FIP addressing mode:

- The FIP addressing mode configured for the OmniSwitch must match the mode used by the ENode and FCF, otherwise the FIP login process and virtual link establishment will fail.

- Disabling FIP snooping on the switch is required before the addressing mode can be changed.

By default, the addressing mode is set to FPMA. To change the FIP addressing mode, use the **fcoe address-mode** command. For example:

```
-> fcoe fip-snooping disable
-> fcoe address-mode spma
-> fcoe fip-snooping enable
```

Enabling/Disabling FCoE Priority Protection

Note. By default, FCoE priority protection is disabled on the switch. When enabled, any non-FCoE traffic *received on any switch port (not just FCoE ports)* that matches the protected FCoE priority value is either dropped or remarked, based on the priority protection action configured for the switch.

The default priority value that is protected is set to 3, which is a common 802.1p value assigned to FCoE traffic. However, this value is not protected until priority protection is enabled for the switch using the **fcoe priority-protection** command. For example:

```
-> fcoe priority-protection enable
```

Changing the Priority Protection Action

By default, non-FCoE traffic that matches the protected priority value is dropped. However, using the **fcoe priority-protection action** command, it is possible to mark the traffic with a different priority value instead of dropping the traffic. For example, the following command will mark non-FCoE traffic that matches the protected priority value (set to 3 for this example) to priority 7:

```
-> fcoe priority-protection action remark 7
```

Changing the FCoE Protected Priority Value

The FCoE priority value specifies an 802.1p value for FCoE traffic that is protected when FCoE priority protection is enabled for the switch. When non-FCoE traffic received on *any* switch port is marked with a priority value that matches the FCoE priority *and* priority protection is enabled, the non-FCoE traffic is dropped or re-marked with a different priority value.

By default, the 802.1p priority value for FCoE traffic is set to 3. To change the FCoE priority value, use the **fcoe priority** command. For example:

```
-> fcoe priority 7
```

In addition, it is also possible to configure two priority values for FCoE traffic. For example, the following command specifies priority 2 and 7:

```
-> fcoe priority 2 7
```

Each time a priority value is configured, the existing priority value is overwritten.

If two priority values are configured but there is a need to change only one of the values, both priority values must be specified with the **fcoe priority** command. For example, if the current priority is set to 2 and 5, to change priority 2 to 3, specify both 3 and 5 as the priority values.

```
-> fcoe priority 3 5
```

Make sure that the FCoE protected priority value is the same priority value that is designated for FCoE traffic in the network configuration.

Configuring the Filtering Resource Threshold

The filtering resource threshold specifies a percentage of switch resources used by FCoE ACLs that when reached, will generate an SNMP trap. By default, this threshold is set to 80%. To change the threshold value, use the **fcoe filtering-resource trap-threshold** command. For example:

```
-> fcoe filtering-resource trap-threshold 50
```

To disable trap generation in this case, set the threshold value to zero. For example:

```
-> fcoe filtering-resource trap-threshold 0
```

Configuring the Keepalive Waiting Time

ENodes and FCFs will send keepalive messages to verify the reachability of the ENode or FCF. These messages will pass through a FIP snooping OmniSwitch that is part of a multi-hop FCoE network. The OmniSwitch will wait a configurable amount of time for the next keepalive message. If no message is received before that time is reached, the switch will clear the FCoE session information.

By default, the amount of time the switch will wait for keepalive messages is set to 300 seconds. To change this amount of time, use the **fcoe house-keeping-time-period** command. For example:

```
-> fcoe house-keeping-time-period 120
```

To disable the house keeping timer, set the timer value to zero. For example:

```
-> fcoe house-keeping-time-period 0
```

Configuring FCoE VLANs

The OmniSwitch implementation of FIP snooping is deployed using FCoE VLANs. Configuring an FCoE VLAN on each transit OmniSwitch between ENodes and FCF is required. Ports or link aggregates (FCoE interfaces) tagged with an FCoE VLAN will carry the FIP control frames and FCoE encapsulated frames through the Ethernet network.

To configure an FCoE VLAN on the OmniSwitch, use the **fcoe vlan** command. For example:

```
-> fcoe vlan 500
```

When creating an FCoE VLAN, specify a VLAN ID that does not exist in the switch configuration, or specify the ID of a dynamically created MVRP VLAN.

By default, an FCoE VLAN is administratively enabled when the VLAN is created. To disable the FCoE VLAN, use the **fcoe vlan** command with the **disable** option:

```
-> fcoe vlan 500 disable
```

To enable a disabled FCoE VLAN, use the fcoe vlan command with the enable option:

```
-> fcoe vlan 500 enable
```

When an FCoE VLAN is created, it is possible to assign an ASCII text name to the VLAN using the **fcoe vlan** command with the **name** option:

```
-> fcoe vlan 500 name fcoe-vlan1
```

If a name is not assigned to an FCoE VLAN when the VLAN is created, the VLAN ID is used by default. For example, when FCoE VLAN 500 is created without a name, then "VLAN-500" is used for the name.

Configuring the FCF MAC Address

Virtual FC links that traverse the lossless Ethernet network through transit switches send FCoE frames to and from the FCF MAC address. This address is learned through the FIP FCF discovery process, but it is possible to statically configure an FCF MAC address for an FCoE VLAN.

By default, the learned FCF MAC address is used. To statically configure an FCF MAC for an FCoE VLAN, use the **fcoe fcf mac** command and specify an existing FCoE VLAN ID. For example:

```
-> fcoe fcf mac 30:10:94:01:00:00 vlan 100
```

To remove a statically configured FCF MAC address from an FCoE VLAN, use the **no** form of the **fcoe fcf mac** command. For example:

```
-> no fcoe fcf 30:10:94:01:00:00 vlan 100
```

Configuring the FC-MAP Prefix

When the FIP addressing mode is set to FPMA, the FCF assigns a 48-bit address to ENode VN_Ports. This address consists of the 24-bit FCoE mapped address prefix (FC-MAP) combined with a 24-bit FC identifier (FCID). The FC-MAP is a configurable value on FCoE VLANs.

The configured FC-MAP value assigned to an FCoE VLAN must match the FC-MAP value used by the FCF device. FCF advertisement packets that contain a different FC-MAP are not processed by the switch

By default, the FC-MAP is set to 0E:FC:00. To statically configure an FC-MAP for an FCoE VLAN, use **fcoe fc-map** command and specify an existing FCoE VLAN ID. For example:

```
-> fcoe fc-map 0E:FC:04 vlan 30
```

To remove a statically configured FC-MAP address from an FCoE VLAN, use the **no** form of the **fcoe fc-map** command. For example:

```
-> no fcoe fc-map 0E:FC:04 vlan 30
```

Configuring FCoE Ports

FCoE ports will carry the FCoE data frames (encapsulated FC frames) and the FIP control frames through the physical OmniSwitch transit switch path between ENodes and FCFs. In addition, dynamic ACLs generated by OmniSwitch QoS are applied to FCoE ports to monitor and secure FIP communications.

The **fcoe role** command is used to designate a switch port or link aggregate as an FCoE port and define the role the port will play in the FCoE transit switch path. The type of role assigned determines the ACL entries the switch generates to apply to the FCoE port (see "FCoE Port Roles" on page 6-7). For example, the following commands configure a port as an FCoE edge port and a link aggregate as an FCF only port:

```
-> fcoe port 1/8/1 role edge
-> fcoe linkagg 10 role fcf-only
```

In these examples, port 1/8/1 is configured as an FCoE edge port because the port will connect the switch to an FCoE-capable server (CNA with FCoE capability). Link aggregate 10 is configured as an FCoE FCF only port because the aggregate will process ingress traffic from an FCF that is destined for an ENode.

To change the role of an FCoE port, first remove the FCoE configuration from the port, then configure FCoE and the new role again for that same port. For example:

```
-> no fcoe port 1/8/1 role edge
-> fcoe port 1/8/1 role enode-only
```

Configuration Example

FIP snooping provides a mechanism for securing FCoE traffic to facilitate lossless storage convergence. The OmniSwitch supports FIP snooping through the use of dynamic ACLs that ensure that traffic only flows between valid ENodes and FCFs.

The FIP snooping capability along with the support of DCB protocols (PFC, ETS, and DCBx) allow the OmniSwitch to participate in a multi-hop FCoE network as an FCoE transit switch. When FIP snooping is enabled, the switch will inspect FIP frames received on FCoE ports that are assigned to FCoE VLANs. OmniSwitch QoS will then dynamically create the necessary ACLs based on the information in the inspected frames.

The following diagram shows a sample FCoE topology in which FIP snooping OmniSwitches are operating as FCoE transit switches to secure and forward FCoE traffic between FCoE endpoints (ENodes and FCF switch). To the ENodes and FCF, the underlying Ethernet links through the transit switch path appear as a direct point-to-point connection that is expected in a native FC network.
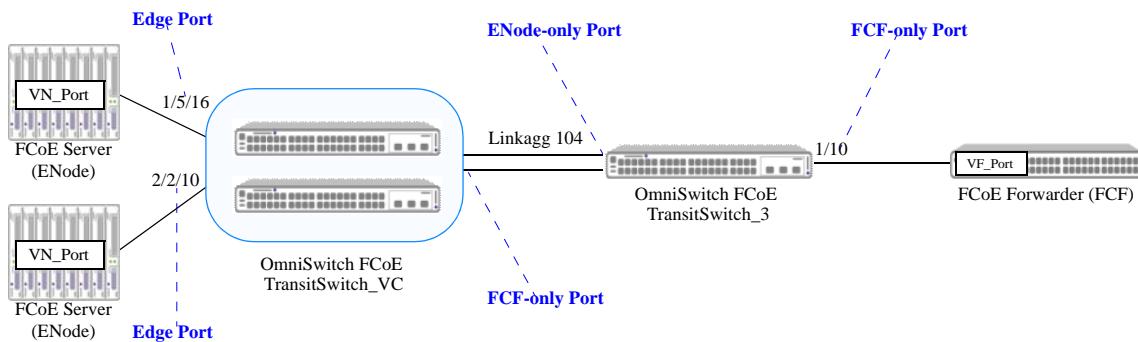
Figure 7: Sample FCoE Topology

The following switch configuration information provides an example of how the transit OmniSwitches are configured in the sample FCoE topology.

- ENodes are connected to ports 1/5/16 and 2/2/10 on TransitSwitch_VC; both ports are configured as FCoE edge ports because they directly connect to a server ENode (FCoE-capable CNA). For example:

  ```
  TransitSwitch_VC-> fcoe port 1/5/16 role edge
  TransitSwitch_VC-> fcoe port 2/2/10 role edge
  ```

- The FCF is connected to port 1/10 on TransitSwitch_3; port 1/10 is configured as an FCF-only port. For example:

  ```
  TransitSwitch_3-> fcoe port 1/10 role fcf-only
  ```

- Linkagg 104 ports are spread across switch modules on both the master and slave chassis of TransitSwitch_VC; linkagg 104 is configured as an FCF-only port on TransitSwitch_VC and as an ENode-only port on TransitSwitch_3. For example:

  ```
  TransitSwitch_VC-> fcoe linkagg 104 role fcf-only
  TransitSwitch_3-> fcoe linkagg 104 role enode-only
  ```

- FCoE VLAN 46 is configured across the transit switch path (manually on each OmniSwitch and, if necessary, manually on the ENodes and FCF). For example:

  ```
  TransitSwitch_VC-> fcoe vlan 46
  TransitSwitch_3-> fcoe vlan 46
  ```

- All FCoE ports and the link aggregate ID are assigned (untagged) to default VLAN 200. The default VLAN carries the ENode VLAN discovery messages; all other FCoE traffic is forwarded on FCoE VLAN 46. For example:

```
TransitSwitch_VC-> vlan 200
TransitSwitch_VC-> vlan 200 members port 1/5/16 untagged
TransitSwitch_VC-> vlan 200 members port 2/1/10 untagged
TransitSwitch_VC-> vlan 200 members linkagg 104 untagged

TransitSwitch_3-> vlan 200
TransitSwitch_3-> vlan 200 members port 1/10 untagged
TransitSwitch_3-> vlan 200 members linkagg 104 untagged
```

- All FCoE ports and the link aggregate ID are tagged with FCoE VLAN 46. For example:

```
TransitSwitch_VC-> vlan 46 members port 1/5/16 tagged
TransitSwitch_VC-> vlan 46 members port 2/1/10 tagged
TransitSwitch_VC-> vlan 46 members linkagg 104 tagged

TransitSwitch_3-> vlan 46 members port 1/10 tagged
TransitSwitch_3-> vlan 46 members linkagg 104 tagged
```

- All FCoE ports are QoS trusted with the default classification set to 802.1p. For example:

```
TransitSwitch_VC-> qos port 1/5/16 trusted default classification 802.1p
TransitSwitch_VC-> qos port 2/1/10 trusted default classification 802.1p
TransitSwitch_VC-> qos linkagg 104 trusted default classification 802.1p

TransitSwitch_3-> qos port 1/10 trusted default classification 802.1p
TransitSwitch_3-> qos linkagg 104 trusted default classification 802.1p
```

- Pre-defined DCB profile 9 (DCP-9) is assigned to all FCoE ports; DCP-9 configures lossless priority flow control for all CoS priorities. For example:

```
TransitSwitch_VC-> qos qsi port 1/5/16 qsp dcb dcp-9
TransitSwitch_VC-> qos qsi port 2/1/10 qsp dcb dcp-9
TransitSwitch_VC-> qos qsi linkagg 104 qsp dcb dcp-9

TransitSwitch_3-> qos qsi port 1/10 qsp dcb dcp-9
TransitSwitch_3-> qos qsi linkagg 104 qsp dcb dcp-9
```

In the sample FCoE topology, FCoE traffic is marked with priority 3.

- The LLDP Application Priority TLV is configured for FCoE priority 3 on FCoE ports 1/5/16, 2/2/10, and 1/10 to allow DCBx (enabled by default) to negotiate Ethernet capabilities and advertise the FCoE priority to the ENodes and FCF. For example:

```
TransitSwitch_VC-> lldp port 1/5/16 tlv application enable
TransitSwitch_VC-> lldp port 1/1/16 tlv application fcoe priority 3
TransitSwitch_VC-> lldp port 2/1/10 tlv application enable
TransitSwitch_VC-> lldp port 2/1/10 tlv application fcoe priority 3

TransitSwitch_3-> lldp port 1/10 tlv application enable
TransitSwitch_3-> lldp port 1/10 tlv application fcoe priority 3
```

- The FIP snooping functionality is globally enabled for all FCoE transit OmniSwitches. When FIP snooping is enabled, FCoE traffic is dropped on all switch VLANs and ports that are *not* configured as FCoE VLANs and ports. For example:

```
TransitSwitch_VC-> fcoe fip-snooping admin-state enable
TransitSwitch_3-> fcoe fip-snooping admin-state enable
```

Verifying the FIP Snooping Configuration

Displaying the FIP Snooping configuration is helpful to verify the actual configuration and monitor FIP snooping sessions on each OmniSwitch that will operate as an FCoE transit switch. To display information about the FIP snooping configuration, use the **show** commands listed in this section.

show fcoe	Displays the global FCoE FIP snooping status and configuration for the switch.
show fcoe ports	Displays the FCoE port configuration, including port roles, for the switch. The port role determines the FIP snooping ACL configuration for the port.
show fcoe sessions	Displays the FIP sessions associated with the local switch.
show fcoe enode	Displays ENode information for the FIP sessions associated with the local switch.
show fcoe fcf	Displays FCF information for the FIP sessions associated with the local switch.
show fcoe fc-map	Displays the Fibre Channel Mapped Address Prefix (FC-MAP) configuration.
show fcoe statistics	Displays ENode and FCF generated session statistics. Use the **clear fcoe statistics** command to reset the statistics information.

7 Configuring an FCoE/FC Gateway

The OmniSwitch provides Fibre Channel over Ethernet (FCoE) convergence solutions that facilitate the expansion of a Fibre Chanel (FC) storage area network (SAN) across an existing Ethernet infrastructure, without having to purchase or manage additional, costly FC equipment. FCoE convergence features supported include the following:

- **FCoE transit switch**—The OmniSwitch supports the FCoE technology used to tunnel FC frames encapsulated within Ethernet MAC frames. To provide the necessary FCoE transit switch functionality, the OmniSwitch supports FCoE Initialization Protocol (FIP) snooping and Data Center Bridging (DCB) protocols for lossless Ethernet. A transit switch is basically a Layer 2 DCB switch that bridges encapsulated FCoE traffic over the Ethernet fabric between FCoE end devices.

- **FCoE/FC gateway switch**—The OmniSwitch serves as an FCoE forwarder to connect FCoE nodes to FC switches, connect FC nodes to an FCoE forwarder, and connect native FC fabrics across an FCoE network. To provide the necessary FCoE/FC gateway functionality, the OmniSwitch supports the following operational modes:

  – N_Port proxy operation to aggregate FCoE Node (ENode) logins over a single OmniSwitch FC port that is connected to an FC switch.

  – F_Port proxy operation to connect FC nodes to an FCoE forwarder or another gateway switch through an FCoE network or on the same gateway switch.

  – E_Port proxy operation to provide a transparent point-to-point FC link between native E_Ports. This allows inter-switch link (ISL) tunneling between FC fabrics over an FCoE network.

An OmniSwitch FCoE transit switch can connect to an OmniSwitch FCoE/FC gateway to access the necessary gateway services needed to transport FCoE traffic to or from the FC SAN. An OmniSwitch FCoE/FC gateway runs FIP snooping on the 10G Ethernet FCoE ports that connect to an FCoE network. On the same switch, FC ports connect to native FC switches or nodes. Traffic is transmitted between the FCoE network and the FC SAN through the gateway switch.

In This Chapter

This chapter describes the OmniSwitch FCoE/FC gateway functionality in general and how the components of this functionality are applied to the switch. It provides information about configuring FCoE/FC gateway components through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following topics and procedures are included in this chapter:

The acronyms and abbreviations used in this chapter are defined here:

CNA	Converged Network Adapter
CVL	Clear Virtual Link
E2E	E_Port-to-E_Port Tunnel
ELP	Exchange Link Parameters
ENode	FCoE Node
E_Port	Expansion Port
F_Port	Fabric Port
FC	Fibre Channel
FCF	FCoE Forwarder
FDISC	Fabric Discovery
FCID	Fabric Port ID (same as N_Port ID).
FCoE	Fibre Channel over Ethernet
FIP	FCoE Initialization Protocol
FLOGI	Fabric Login
FLOGO	Fabric Logout
HBA	Host Bus Adapter
ISL	Inter-switch Link
NPIV	N_Port ID Virtualization
N_Port	Node Port
NP_Port	Proxy Node Port

TE_Port	Tunnel Expansion Port
VE_Port	Virtual E_Port
VF_Port	Virtual F_Port
VN_Port	Virtual N_Port
VSAN	Virtual Storage Area Network
WWNN	World Wide Node Name
WWPN	World Wide Port Name

For more information about the OmniSwitch FIP snooping implementation, see Chapter 6, "Configuring FIP Snooping."

For more information about OmniSwitch support for lossless Ethernet, see Chapter 2, "Configuring Data Center Bridging."

FCoE/FC Gateway Overview

The OmniSwitch implementation of FCoE/FC gateway functionality allows the switch to transparently connect FCoE and FC nodes with an FC SAN across an FCoE (lossless Ethernet) network. To provide this type of connectivity, an OmniSwitch FCoE/FC gateway supports the following three modes of operation that are used to converge FC over Ethernet and FC-to-FC over Ethernet:

- **N_Port Proxy mode**—allows ENodes in an FCoE network and FC switches in an FC SAN to communicate with each other. To an ENode the OmniSwitch gateway emulates an FCoE forwarder; to an FC switch the OmniSwitch gateway emulates an N_Port ID Virtualization (NPIV) host. See "Using the N_Port Proxy Mode" on page 7-6 and "Configuring an N_Port Proxy Operation" on page 7-22 for more information.

- **F_Port Proxy mode**—allows FC nodes to connect with FC switches and FCFs across an FCoE network. The OmniSwitch gateway forwards login requests from an FC node (N_Port on a server or storage with an HBA) across Ethernet via an FCoE VLAN to an NPIV node or FCF. See "Using the F_Port Proxy Mode" on page 7-11 and "Configuring an F_Port Proxy Operation" on page 7-24 for more information.

- **E_Port Proxy mode**—allows FC switches to set up inter-switch link trunking between FC fabrics over an FCoE network. The OmniSwitch gateway provides an E_Port to E_Port (E2E) tunneling function that emulates a point-to-point FC link between E_Ports on native FC switches. See "Using the E_Port Proxy Mode" on page 7-14 and "Configuring an E_Port Proxy Operation" on page 7-25 for more information.

The OmniSwitch FCoE/FC gateway sits at the entry point of an FC fabric, which is required to handle the login process for ENodes and FC nodes accessing the fabric through the gateway switch.

OmniSwitch FCoE/FC Gateway Fabric

The OmniSwitch FCoE/FC gateway fabric identifies the ports on which FC traffic is converted to FCoE traffic and FCoE traffic is converted to FC traffic. The fabric itself is defined through the association of the FC ports to an OmniSwitch VSAN and the association of FCoE ports to an FCoE VLAN. The VSAN is then mapped to the FCoE VLAN to define the traffic path for converted FC or FCoE frames.

- **VSAN**—carries native FC traffic to and from FC SAN devices accessing the gateway through the OmniSwitch FC ports assigned to the VSAN. Initially all FC ports are assigned to a default VSAN (VSAN 1). Additional VSANs are configurable to define additional gateway fabrics, similar to how VLANs are used to define additional broadcast domains.

- **FCoE VLAN**—carries FCoE traffic to and from FCoE network devices accessing the gateway through OmniSwitch FCoE ports associated with the FCoE VLAN.

- **FCoE ports**—OmniSwitch 10G Ethernet ports configured with a lossless Data Center Bridging profile and an FCoE port role for FIP snooping. Each interface is assigned to an FCoE VLAN.

- **FC ports**—OmniSwitch Fibre Channel ports configured to operate in N_Port proxy mode or F_Port proxy mode. Each FC interface is assigned to a single VSAN, but it is possible to assign multiple FC ports to the same VSAN.

- **VSAN-to-FCoE VLAN mapping**—an association between one OmniSwitch VSAN and one FCoE VLAN.

  – Native FC traffic received on FC ports associated with the mapped VSAN is converted to FCoE traffic (encapsulated FC frames) and forwarded on the FCoE ports associated with the FCoE VLAN.

  – FCoE traffic received on FCoE ports associated with the mapped FCoE VLAN is converted to native FC traffic.

Note. The OmniSwitch FCoE/FC gateway fabric is only used for N_Port proxy and F_Port proxy operations. The E_Port proxy mode does not use the FCoE gateway fabric to establish an E2E tunnel path.

How it Works

Each VSAN-to-FCoE VLAN mapping and associated ports represents an independent fabric on the OmniSwitch gateway. Devices only communicate with other devices connected through the same fabric on the gateway switch. For example, the following diagram shows an OmniSwitch FCoE/FC gateway switch with two separate fabrics: one fabric flows through VSAN 1 and the second fabric flows through VSAN 2.

Figure 1: OmniSwitch FCoE/FC Gateway Fabric

In this example:

- VSAN 1 is associated with FC ports 2/1 and 2/2 and is mapped to FCoE VLAN 200, which is tagged with FCoE ports 1/1 and 1/2.

- VSAN 2 is associated with FC ports 2/3 and 2/4 and is mapped to FCoE VLAN 300, which is tagged with FCoE ports 1/2 and 1/3.

- VSAN 1 FC ports are connected to FC Switch 1; VSAN 2 FC ports are connected to FC Switch 2.

- The OS6900 gateway switch sends out two discovery advertisement messages: one message for VSAN 1 and one message for VSAN 2. *Each active VSAN represents a separate FCoE forwarder.*

- ENode requests with VLAN ID 200 are sent to FC Switch X and ENode requests with VLAN 300 are sent to FC Switch Y.

- Communication between VSANs (fabrics) is not allowed. As a result, VLAN 200 traffic cannot reach FC Switch Y and VLAN 300 traffic cannot reach FC Switch X.

Using the N_Port Proxy Mode

The OmniSwitch FCoE/FC gateway utilizes N_Port virtualization functionality to allow the switch to serve as an N_Port proxy for FCoE nodes (ENodes) accessing an FC SAN over an FCoE network.

- Fabric log in requests and log in accept messages are relayed between ENodes and FC switches through the gateway switch.

- An FC switch connected to the gateway switch is presented with a proxy N_Port (NP_Port) that can request multiple N_Port IDs on the same physical port. To the FC switch, the OmniSwitch gateway is just another NPIV-enabled host. As a result, the FC switch should support NPIV.

- An ENode accessing the FC switch fabric through the gateway switch is presented with a VF_Port upon successful fabric login.

As an N_Port proxy, the OmniSwitch aggregates FCoE node (ENode) fabric login requests and subsequent N_Port sessions through one physical OmniSwitch FC port. Each session is still identified by the WWPN and FCID assigned to the ENode even though the session is part of an aggregate flow.

The FC switch that provides login services through the OmniSwitch NP_Port treats each ENode session as a separate virtual N_Port connection. This helps to conserve on the number of physical ports required to handle ENode traffic, while making it possible to still track and service VMs, users, or applications from one or more ENodes on an individual basis.

The following diagram provides a high-level overview of the NPIV functionality the FCoE/FC Gateway provides when the switch is operating in the N_Port proxy mode:
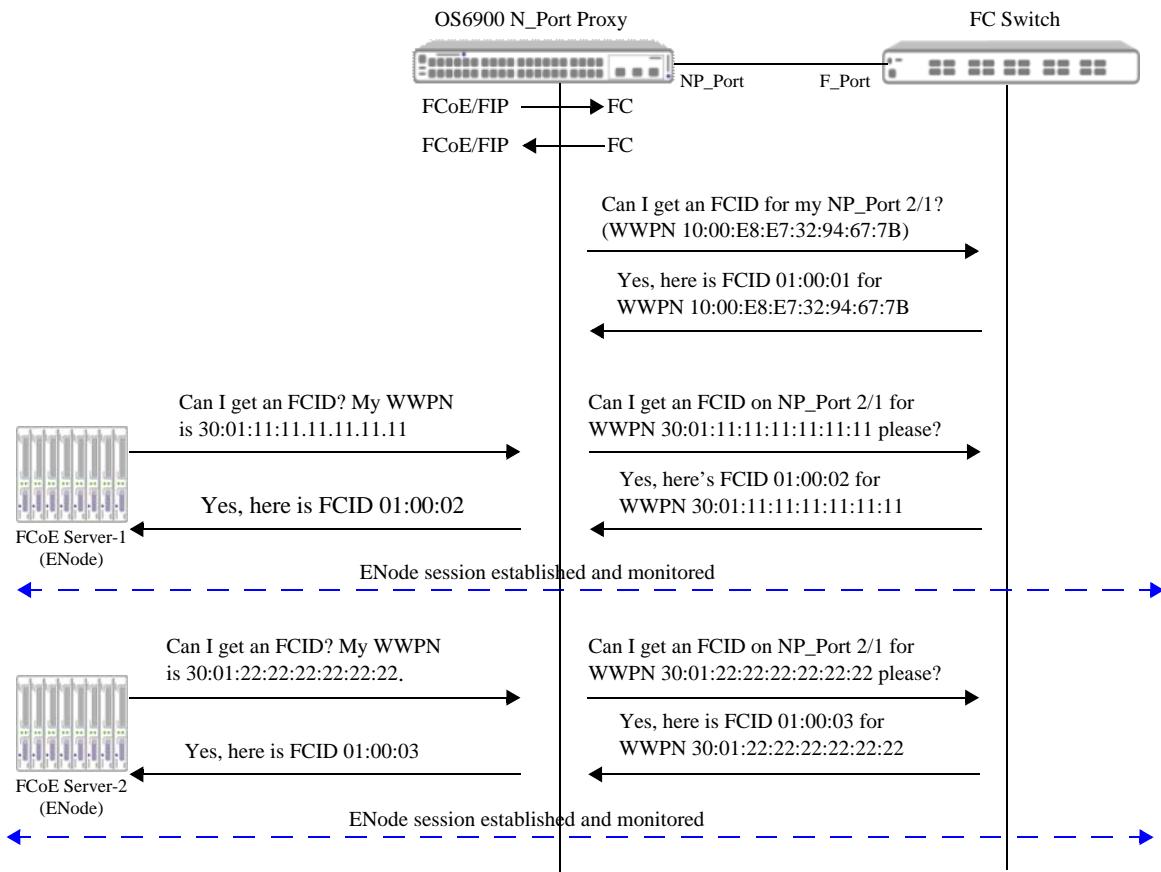
Figure 2: OmniSwitch N_Port Proxy Overview

In this diagram,

1 The OS6900 port connecting to the FC switch is an FC port configured to operate as an NP_Port.

2 When the NP_Port comes up, the port attempts a fabric login through the F_Port on the FC switch.

3 Upon a successful login, the NP_Port is assigned a Fibre Channel ID (also called an N_Port ID).

4 The NP_Port can then request additional N_Port IDs for ENodes over the same physical port.

The N_Port proxy mode is activated on the switch when an FCoE VLAN is mapped to a VSAN, and at least one FC interface assigned to the mapped VSAN is configured as a proxy node port (NP_Port).

The following diagram shows an example network topology in which an OmniSwitch FCoE/FC gateway is operating as an N_Port proxy to allow ENodes to log into an FC SAN:
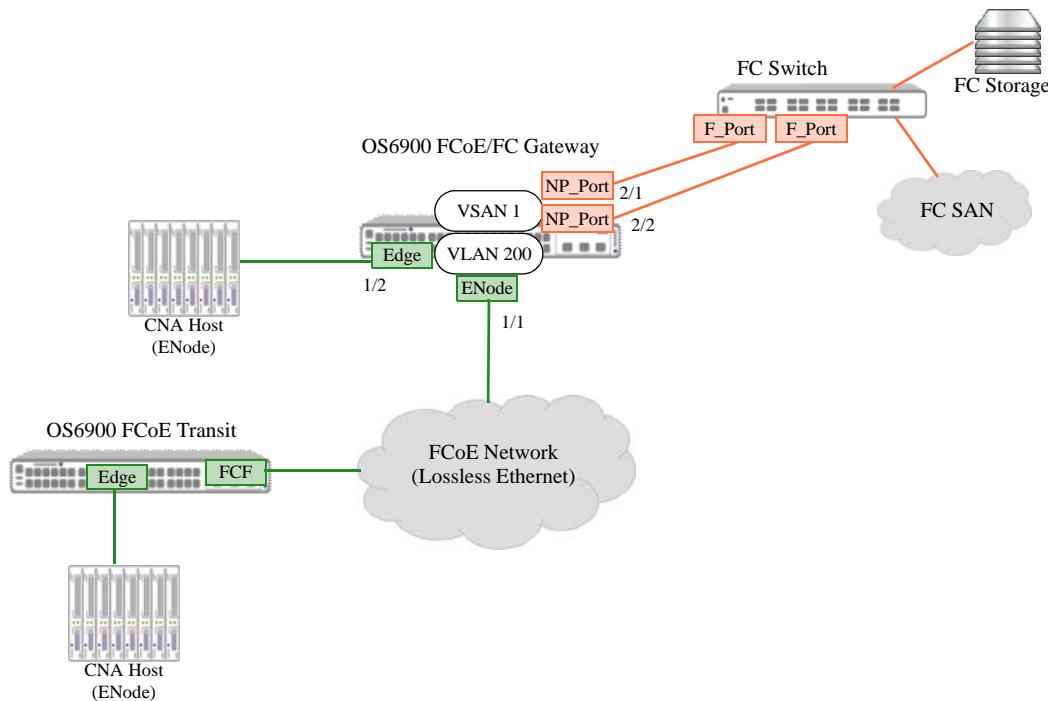
Figure 3: OmniSwitch N_Port Proxy (NPIV) Example

In this example configuration,

- The OmniSwitch FCoE/FC gateway fabric consists of VSAN 1 mapped to FCoE VLAN 200, NP_Ports 2/1 and 2/2 assigned to VSAN 1, and FCoE ports 1/1 and 1/2 tagged with FCoE VLAN 200.

- FCoE VLAN 200 is a VLAN dedicated to carrying FCoE and FIP traffic between ENodes and an FCF through an FCoE network. The OS6900 transit switch is part of the FCoE network on which FCoE VLAN 200 is also configured.

- Port 1/1 is configured as an FCoE ENode-only port; and port 1/2 is configured as an FCoE edge port.
  - If the load balancing method for the switch is set to dynamic or dynamic re-order, ACLs are needed irrespective of port roles. An ACL entry is created for sessions established over FCoE ports configured as edge, ENode-only, mixed, and trusted ports.
  - If the load balancing method for the switch is set to ENode-based or FCoE ports are statically mapped to FC ports, then an ACL entry is created for sessions established over edge ports only (similar to FIP Snooping).

- FCoE traffic received from ENode devices (CNA hosts) on FCoE ports 1/1 and 1/2 is converted to FC traffic and sent out on FC port 2/1 or port 2/2 to the FC switch.

- FC traffic received from the FC switch on NP_Ports 2/1 and 2/2 is converted to FCoE traffic and sent out on FCoE port 1/1 or 1/2.

See for more information about configuring the N_Port proxy mode on an OmniSwitch FCoE/FC gateway.

How it Works

When an OmniSwitch FC port is configured to operate as an NP_Port, the OmniSwitch gateway automatically initiates a fabric login for the NP_Port. If the fabric login is successful, the FC switch assigns a unique Fibre Channel ID (FCID) to the NP_Port. At this point, the N_Port proxy mode is active on the NP_Port.

Once the N_Port proxy mode is active and there is a VSAN-to-FCoE VLAN mapping, the OmniSwitch gateway can provide services to ENodes by participating in FCoE VLAN discovery and FCF discovery by sending discovery advertisement messages.

- When an FCoE VLAN discovery message is received from an ENode, the OmniSwitch gateway will respond back with the FCoE VLAN ID (for example, FCoE VLAN 200 as shown in the diagram on).

- When an FCF discovery solicitation message from an ENode is received, the OmniSwitch gateway will respond back with a unicast discovery advertisement message.

The OmniSwitch FCoE/FC gateway forwards multicast discovery messages initiated by an ENode within the designated FCoE VLAN. If this VLAN also provides access to FCFs or other NPIV gateways, the ENode may receive replies from multiple FCFs or gateways, including the OmniSwitch. The FCF replies contain a FIP priority, a verified maximum FCoE frame size and an availability bit (A-bit) setting. Based on this information, the ENode selects one of the FCFs for FC fabric login (FLOGI).

- If the OmniSwitch gateway is not selected, then the switch will serve as an FCoE transit switch (FIP snooping bridge) and forward the ENode traffic. No N_Port proxy functionality is provided for the ENode.

- If the ENode does select the OmniSwitch gateway, the ENode sends a unicast FIP FLOGI request to the OmniSwitch.

In the example shown on , the OmniSwitch FCoE/FC gateway is the selected FCoE forwarder for both ENodes. When the OmniSwitch receives a FIP FLOGI request with VLAN 200 from an ENode, the OmniSwitch creates an FC fabric discovery (FDISC) message and transmits the message on one of the NP_Ports in VSAN 1.

After the FDISC messages is sent, the OmniSwitch gateway may receive an FC Link Service Accept (LS_ACC) or Link Service Reject (LS_RJT) reply from the FC switch connected to the NP_Port on which the FDISC message was sent.

- If an FC LS_ACC message is received, the OmniSwitch gateway will generate a unique VN_Port MAC address (combination of FC_MAP + FCID), convert the FC LS_ACC message into a FIP LS_ACC message containing the VN_Port MAC address and then forward the LS_ACC to the ENode.

- If an FC LS_RJT message is received from the FC switch, the OmniSwitch gateway will simply convert the message to a FIP LS_RJT message and then forward the message to the ENode.

When the ENode successfully logs into the fabric, the OmniSwitch gateway will dynamically install ACLs (FIP snooping) on the FCoE port where the FLOGI request from the ENode was received. In addition, the OmniSwitch gateway will monitor the established login connections as follows:

- The OmniSwitch will send out periodic multicast discovery advertisements to the ALL-ENODE-MAC to notify ENodes of its availability for new ENodes and its reachability for ENodes with sessions already established through the gateway.

- The OmniSwitch listens for ENode and VN_Port keep-alive messages. If keep-alive messages are not received, the switch will clear all the corresponding sessions.

- There are two types of logout events that will cause the switch to clear all corresponding sessions:
  - An explicit logout event triggered by an ENode, FC switch, or OmniSwitch gateway.
  - An implicit logout event triggered by a link down or missed keep-alive messages.

N_Port Proxy Load Balancing

When more than one FC port is assigned to the same VSAN, the switch will automatically apply a load balancing algorithm to determine on which NP_Port the FC FDISC packet is forwarded to an FC switch. If one of the FC ports goes down, then all sessions on that port are torn down and logged in again on one of the other available FC ports in the same VSAN.

There are four load balancing methods supported: dynamic (the default), dynamic reorder, dynamic ENode based, and static. All three of the load balancing algorithms are dynamic, in that the process used to distribute ENode sessions automatically determines which NP_Ports to use. However, it is also possible to manually configure a static association between an FCoE port and an NP_Port.

The active load balancing algorithm is globally configured and applied on all OmniSwitch NP_Ports; load balancing is not applied on OmniSwitch F_Ports. In addition, the port selection is made based on the incoming ENode FLOGI requests, not on the outgoing FDISC messages for those same requests.

This section provides an overview of each load balancing method, including static FCoE port assignment.

Dynamic Load Balancing

By default, the dynamic load balancing algorithm is used. When this method is applied, the switch selects the NP_Port with the least number of logins. If all the NP_Ports in the VSAN have the same number of logins, then a round-robin selection method is used.

The dynamic load balancing method is only applied to new login requests received; none of the sessions already active on the NP_Ports are disturbed. For example, if there is one NP_Port with active sessions and another NP_Port is added to the same VSAN, the existing sessions on the first NP_Port are left alone. New requests are forwarded on the new NP_Port because initially it has the least number of logins.

Dynamic Reorder Load Balancing

When the dynamic reorder method is selected, load balancing is applied to all ENode login sessions (not just the new ones) to ensure an even distribution of session across all NP_Ports in the same VSAN. This means that the switch might have to tear down some of the existing sessions and re-establish those sessions on a different NP_Port.

For example, if an NP_Port has two active sessions and another NP_Port is added to the same VSAN, the switch explicitly logs out one of the ENode sessions and sends a CVL to the ENode that was logged out. When the ENode receives the CVL, will send another FLOGI request that the switch will now redirect onto the added NP_Port. The other ENode session continues on the first port.

ENode-based Load Balancing

When the ENode-based load balancing method is selected, each NP_Port in the VSAN will send a multicast discovery advertisement to all ENodes in the mapped FCoE VLAN. The FCF MAC address in the discovery advertisement is the MAC address of the FC port. The ENode then decides which NP_Port to use for the login.

Similar to the dynamic load balancing method, the ENode-based method does not disturb existing ENode sessions.

Static FCoE Port Mapping

Static mapping of an FCoE port to an NP_Port ensures that all ENode sessions on the FCoE port are specifically directed to the mapped NP_Port. No other NP_Port is eligible to receive sessions from this FCoE port. For example, if FCoE port 1/1 is mapped to NP_Port 2/1, then all ENode FLOGI requests are automatically sent to NP_Port 2/1. There is no consideration for the number of sessions already on that port, or on any other NP_Port in the VSAN.

Note. Use caution when configuring this method. A static assignment of an FCoE port to an NP_Port overrides dynamic load balancing.

For more information about configuring the four load balancing methods, see "Configuring N_Port Proxy Load Balancing" on page 7-23.

Using the F_Port Proxy Mode

The F_Port proxy functionality allows the OmniSwitch gateway to provide an F_Port fabric to FC nodes (HBA N_Ports). On the FC-facing side of the gateway switch, FC nodes are presented with F_Ports. On the FCoE-facing side of the gateway switch, FCoE forwarders or other NPIV proxy gateways are presented with VN_Ports.

Utilizing the F_Port proxy functionality, an OmniSwitch gateway can perform the following tasks on behalf of FC nodes connected to the gateway in an FCoE network:

- Discovers FCoE forwarders and NPIV proxy gateways on the FCoE network and selects one of these devices to receive converted FLOGI requests from the attached FC nodes.

- Converts FC node FLOGI requests to FIP FLOGI requests and forwards them to the selected FCoE forwarder or NPIV proxy gateway through the FCoE network.

- Converts FIP login accept or reject responses received from the FCoE forwarder to FC responses that are then forwarded to the originating FC node.

- Sends periodic keep-alive messages to the FCoE forwarder to maintain virtual connections between the attached FC nodes and the FC SAN switch.

The F_Port proxy functionality is activated on the switch when FIP snooping is configured and enabled, an FCoE VLAN is mapped to a VSAN, and at least one FC interface assigned to the mapped VSAN is configured as a fabric port (F_Port).

The following diagram shows an example network topology in which an OmniSwitch FCoE/FC gateway is operating as an F_Port proxy to allow FC nodes to log into an FC SAN over an FCoE network:

Figure 4: OmniSwitch F_Port Proxy (Reverse-NPIV) Example

In this example configuration,

- The OS6900-Gateway A switch is operating as an N_Port proxy. The OS6900-Gateway B switch is operating as an F_Port proxy switch.

- The FCoE/FC gateway fabric connects the two gateway switches through the FCoE (lossless Ethernet) network, allowing the FC nodes to access the FC switch.

- The FCoE/FC gateway fabric is configured as follows on each OmniSwitch gateway:
  - VSAN 1 is mapped to FCoE VLAN 200.
  - FC ports are assigned to VSAN 1.
  - FCoE ports are assigned to VLAN 200.

- FCoE VLAN 200 is a VLAN dedicated to carrying FCoE and FIP traffic between the two gateways.

- On the Gateway B switch, FC ports 2/3 and 2/4 are configured to operate in the fabric port (F_Port) mode and are also assigned to VSAN 1. These ports are connected to FC nodes. The Gateway B switch will provide the connected FC nodes with access to FC devices across the FCoE network.

- On the Gateway A switch, NP_Ports 2/1 and 2/2 are assigned to VSAN 1. These ports will provide N_Port proxy access to the FC switch.

- FCoE ports 1/1 and 1/2 are tagged with FCoE VLAN 200.
  - Port 1/1 is configured as an FCoE ENode-only port, which means that dynamic ACLs are applied to ENode traffic received on that port from the R-NPIV proxy switch.
  - Port 1/2 is configured as an FCoE FCF-only port, which means that dynamic ACLs are applied to traffic received directly from the NPIV proxy switch.

- FC traffic received from the FC nodes on F_Ports 2/3 and 2/4 is converted to FCoE traffic and sent out FCoE port 1/2 to the Gateway A switch.

- FCoE traffic received from the Gateway B switch on FCoE port 1/1 is converted to FC traffic and sent out on NP_Ports 2/1 or 2/2 to the FC switch.

See "Configuring an F_Port Proxy Operation" on page 7-24 for more information about configuring the F_Port proxy mode on an OmniSwitch FCoE/FC gateway.

How it Works

This subsection describes how the OmniSwitch gateway provides F_Port proxy services that connect FC node devices with FC switches across an FCoE network. The process described is based on the example shown in "Figure 4: OmniSwitch F_Port Proxy (Reverse-NPIV) Example" on page 7-12.

- When the FC ports 2/3 and 2/4 are configured to operate as F_Ports and are up and running, the F_Port proxy functionality is implicitly enabled on those ports. One of the main functions of the F_Port proxy switch is to find FCoE forwarders and/or NPIV gateways on behalf of the FC nodes connected to the F_Ports.

- The FCoE/FIP snooping functionality on the OmniSwitch gateway allows the switch to participate in the FIP FCF discovery process to locate FCoE forwarders and NPIV gateway switches in the FCoE network. In the example, the F_Port proxy switch listens on FCoE VLAN 200 for FIP FCF discovery advertisements and finds the OS6900 N_Port proxy gateway.

- The F_Port proxy switch uses FCoE VLAN 200 for FIP FCF discovery because this VLAN is mapped to VSAN 1 as part of the FCoE/FC gateway fabric. Therefore, the F_Port proxy switch did not have to participate in the FIP VLAN discovery process to find VLAN 200 on which to transport FCoE traffic.

- If the F_Port proxy switch found more than one compatible FCF or gateway, the switch maintains a list of these devices by continuing to monitor discovery advertisements. If the selected FCF or gateway switch goes down, another device is selected from the list and all sessions are torn down and instantiated again with the new FCF or gateway.

- When the FC nodes connected to F_Ports 2/3 and 2/4 attempt to log in to the FC fabric by initiating FC FLOGI requests, the F_Port proxy switch converts the requests to FIP FLOGI then forwards the FIP FLOGI requests on FCoE VLAN 200 to the N_Port proxy switch.

- If an FC node sends FDISC messages from an NPIV HBA, these messages are converted in the same manner as the FLOGI messages are converted for transport to and from the FC fabric.

- The N_Port proxy switch converts the FIP FLOGIs received to FC FLOGIs and relays the FC FLOGIs on VSAN 1 to the FC switch. The FC switch then processes the log in request and responds with an FC accept (LS_ACC) or reject (LS_RJT) message.

- When the N_Port proxy switch receives a response from the FC switch, the gateway converts the response to a FIP response and forwards the response on FCoE VLAN 200 to the F_Port proxy switch. The F_Port proxy switch then converts the response back to an FC frame and forwards the response on VSAN 1 to the appropriate FC node.

- If the FC node login was accepted and a session established, the F_Port proxy switch maintains the virtual connection by sending periodic keep-alive messages to the N_Port proxy switch on behalf of the FC nodes. This is because an HBA on an FC node does not maintain keep-alive messages.

- If the F_Port proxy switch determines the N_Port proxy switch is no longer reachable, virtual connections established with that switch are torn down. The FC nodes can then initiate login requests again and the F_Port proxy will select another FCoE forwarder or NPIV gateway to establish new virtual connections.

Using the E_Port Proxy Mode

E_Port-to- E_Port (E2E) tunneling provides a method for transparently connecting native FC switches over an FCoE network. The OmniSwitch FCoE/FC gateway serves as an E_Port proxy through the configuration of tunnel expansion ports (TE_Ports) and virtual expansion ports (VE_Ports) to provide connectivity between E_Ports on FC switches over an FCoE network.

There are two ways to establish an E_Port proxy operation to provide an E2E tunnel between two FC switch E_Ports:

- **VE_Port tunneling between two switches**—the tunnel is established through an FCoE network between a VE_Port on one switch and a VE_Port on another switch. Each VE port is part of a tunnel endpoint that is mapped to an OmniSwitch TE_Port via an FCoE VLAN (both port types are associated with the same FCoE VLAN). The TE_Port connects directly to an FC switch E_Port.

- **TE_Port tunneling on the same switch**—an internal tunnel is established between two OmniSwitch TE_Ports on the same switch (or in a virtual chassis configuration). All tunnel traffic is switched across the internal tunnel, instead of over an FCoE network.

The following diagram shows an example of an OmniSwitch VE_Port tunnel configuration:

Figure 5: OmniSwitch E_Port Proxy (E2E Tunnel) Example

In this example configuration,

- An E2E tunnel connects the two FC SANs through the FCoE network.

- FCoE VLAN 500 is configured on both switches and identifies the tunnel path through the FCoE network.

- OS6900 10G Ethernet ports 1/1 and 1/2 are configured as FCoE VE_Ports.

- OS6900 FC ports 2/1 and 2/2 are configured as tunnel expansion ports (TE_Ports) and connect to an E_Port on an FC switch.

- The E2E tunnel path is defined by associating TE_Port 2/1 and VE_Port 1/1 with FCoE VLAN 500 on Gateway A and associating TE_Port 2/2 and VE_Port 1/2 with FCoE VLAN 500 on Gateway B.

- Traffic from each FC SAN is tunneled to and from the OS6900 FC TE_Ports through the FCoE VE_Ports. Each FC switch thinks it is directly connected to the other FC switch; the underlying tunnel through the FCoE network is transparent.

See "Configuring an E_Port Proxy Operation" on page 7-25 for more information about configuring the E_Port proxy mode on an OmniSwitch FCoE/FC gateway.

How it Works

In an FC SAN, the FC switches provide connectivity for nodes accessing devices in the SAN. FC switches are connected to each other through ports designated as expansion ports (E_Ports). This type of port allows two FC switches to set up an inter-switch link (ISL) between each other.

In an FCoE network virtual expansion ports (VE_Ports) provide connectivity between FCoE forwarders or gateways, similar to E_Port connectivity between native FC switches. This type of connectivity provides the ability to extend E_Port connections across an FCoE network by tunneling E_Port traffic between VE_Ports.

On the OmniSwitch, FCoE ports that will tunnel E_Port traffic through the FCoE network, are configured as VE_Ports. An OmniSwitch FC port that will connect directly to an E_Port on an FC switch is configured as a tunnel expansion port (TE_Port). These two port types and a common FCoE VLAN are used to define the E2E tunnel endpoints.

As shown in the Figure 5 example page 7-15, two OmniSwitch FCoE/FC gateways are configured with a single VE_Port and FCoE VLAN 500. In addition, both switches have an FC TE_Port connected directly to an FC switch. On each OmniSwitch, an E2E tunnel endpoint is defined by the association of the VE_Port and the TE_Port to the FCoE VLAN 500.

Defining the E2E tunnel endpoints does not establish a tunnel session. Once the required tunnel components are configured and active, successful exchange of exchange link parameters (ELP) between a TE_Port and a VE_Port will establish the tunnel session.

- VE_Ports participate in the FIP FCF discovery process. In the example, the VE_Ports discover the other OS6900 gateway switch on FCoE VLAN 500.

- The OS6900 gateway will convert ELP received on the TE_Port to FIP ELP and relay the FIP ELP to the selected FCF.

- The receiving VE_Port will validate the FIP ELP request, remove the Ethernet header, and send the FC ELP request to the corresponding TE_Port.

- If the ELP exchange is successful, the tunnel session is instantiated on the VE_Ports.

Once the tunnel session is established, the VE_Ports will maintain the link by sending periodic FIP discovery messages. At this point, all ISL data traffic is allowed to pass through the tunnel.

FIP Packet Processing

The following table shows how the OmniSwitch FCoE/FC gateway switch identifies FIP snooping traffic (FCoE packets) received on FCoE ports to determine which gateway function will process the traffic:

Packet Type	Verifies ...	Assigns Packet to ...
MDS (multicast discovery solicitation)	FCF (F) bit setting	E_Port proxy if the F bit is set to one.
		N_Port proxy if the F bit is set to zero.
UDS (unicast discovery solicitation)	Unicast destination MAC	F_Port proxy if destination MAC is an FC F_Port.
		E_Port proxy if destination MAC is an FC TE_Port.
MDA (multicast discovery advertisement)	Multicast destination MAC	E_Port proxy if the destination MAC is ALL- FCF-MAC.
		F_Port proxy if the destination MAC is ALL- ENODE- MAC.
CVL (clear virtual link)	Unicast destination MAC	F_Port proxy if the destination MAC is an FC F_Port.
		E_Port proxy if the destination MAC is an FC TE_Port.
FIP FLOGI / FIP FDISC	Packet type	N_Port proxy
FIP FLOGO (fabric logout)	Packet type	N_Port proxy
FLOGI/FIDSC LS_ACC FLOGI/FDISC LS_RJCT FLOGO LS_ACC FLOGO LS_RJCT	Packet type	F_Port proxy
VLAN Discovery	F-bit setting	E_Port proxy if the F-bit is set to one.
		N_Port proxy if the F-bit is set to zero.
FIP ELP SW_ACC SW_RJCT	Packet type	E_Port proxy
Keep-alive	Packet type	N_Port proxy

Interaction with Other Features

This section contains important information about how other OmniSwitch features interact with the FCoE/FC gateway features. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

FIP Snooping

The OmniSwitch FCoE/FC gateway functionality is a superset of the OmniSwitch FIP snooping implementation. FIP snooping ensures the security of the FCoE traffic entering the gateway switch. The gateway switch converts FCoE encapsulated frames to FC frames for forwarding onto an FC SAN; the gateway also converts FC frames to FCoE frames for forwarding onto an FCoE network.

Configuring the following FIP snooping components is *required* to support and define the FCoE/FC gateway fabric on the OmniSwitch:

- **FCoE VLAN**—A type of VLAN that is dedicated to carrying only FCoE and FIP traffic. ENodes use this type of VLAN to transmit and receive FCoE and FIP traffic and establish virtual links to an FC SAN through the NPIV gateway.

  Manual configuration of the FCoE VLAN is required along the FCoE network path and on the gateway switch. However, the ENode may invoke FIP VLAN discovery to discover the FCoE VLANs in the network. If not, manual configuration of the FCoE VLANs may also be required on the appropriate ENodes

  If there is no FCoE/FC gateway functionality mapped to an FCoE VLAN, then only FIP snooping is applied to traffic forwarded on the VLAN even if the VLAN is configured on an FCoE/FC gateway switch.

- **FCoE Port Roles**—Determines the types of ACLs that are dynamically generated and applied to FCoE traffic that ingresses on the FCoE port. For the FCoE/FC gateway E2E tunneling feature, the virtual expansion port (VE_Port) role is used to allow exchange link parameters (ELP) to travel between FC switches through an FCoE network. The mixed port role is used when the FCoE port will carry a mixture of traffic (for example, E2E tunnel, ENode, FCF only traffic) all on the same port.

Having a good understanding of FCoE and FIP snooping is highly recommended when attempting to configure FCoE/FC gateway features. For more information, see Chapter 6, "Configuring FIP Snooping."

Data Center Bridging (DCB)

FCoE requires an underlying lossless Ethernet network. The OmniSwitch supports the following Data Center Bridging (DCB) protocols that extend Ethernet capabilities to support the convergence of storage and data in virtualized networks:

- **Priority-Based Flow Control (PFC)**—Based on the IEEE 802.1Qbb standard, PFC pauses traffic based on congestion priority instead of blocking the entire link when congestion occurs. Allows lossless and lossy traffic with different priorities on the same physical port.

- **Enhanced Transmission Selection (ETS)**—Based on the IEEE 802.1Qaz standard, ETS provides a common framework for dynamic bandwidth management. ETS groups related traffic into priority groups (Traffic Classes) to which bandwidth guarantees and scheduling are applied.

- **Data Center Bridging Exchange (DCBX)**—Based on the IEEE 802.1Qaz standard, DCBX uses the Link Layer Discovery Protocol (LLDP) to exchange and negotiate PFC and ETS information between

two directly connected peer switches. Enabled by default, DCBX is responsible for auto-negotiation and auto-configuration of link parameters for DCB functions.

Note. A custom DCB profile for each of the FC ports is required to ensure an end-to-end lossless path through the FCoE/FC gateway. See "Configuring FC Ports" on page 7-21 for more information.

For more information about DCB see Chapter 2, "Configuring Data Center Bridging,"

Hardware

This implementation of OmniSwitch FCoE/FC gateway functionality requires support for both 10G Ethernet and Fibre Channel ports. The OmniSwitch 6900 provides both with the addition of the OS-XNI-U12E module with the SFP-FC-SR transceiver.

The Fibre Channel ports are not available for use until they are configured to operate in a specific Fibre Channel mode (NP_Port, F_Port, or TE_Port). Once configured, each port is assigned a 64-bit world wide port name (WWPN) that is comprised of "10:00" plus the MAC address of the port. For example, 10:00:e8:e7:32:94:67:7d.

The World-wide Node Name (WWNN) for the OmniSwitch 6900 is comprised of "10:00" plus the next available increment of the switch base MAC address. For example, if the switch MAC address is e8:e7:32:6c:4a:39, then the WWNN for the switch is 10:00: e8:e7:32:6c:4a:3a.

FCoE/FC Gateway Configuration Guidelines

OmniSwitch FCoE/FC gateway operations are not automatically activated for the switch; there is no single command to enable or disable gateway functionality. Instead, the configuration of the following software components enables one or more of the supported gateway operations:

- **FIP snooping**—FCoE/FC gateway functionality requires an active FIP snooping configuration. FIP Snooping ensures the security of an FCoE network.

- **Virtual Storage Area Network (VSAN)**—an FC port is assigned to a VSAN to create an NP_Port or F_Port connection to an FC switch or node in that VSAN. Not required for E2E Tunnel configuration.

- **VSAN-to-FCoE VLAN mapping**—identifies the FCoE/FC gateway fabric for the ENode or FC node login process via the FCoE VLAN. Not required for E_Port proxy (E2E Tunnel) configuration. See "OmniSwitch FCoE/FC Gateway Fabric" on page 7-4 for more information.

- **FCoE port role**—an FCoE port serves as an E2E tunnel endpoint in an FCoE network only when the port is configured as a virtual E_Port (VE_Port). The role of other FCoE ports is configured based on the FIP snooping configuration for the gateway switch.

- **FC port mode**—the operational mode of the FC port determines the type of gateway functionality provided on that port. There are three modes supported: N_Port proxy, F_Port proxy, and E_Port proxy. See "Configuring FC Ports" on page 7-21 for more information.

Consider the following general guidelines when configuring an OmniSwitch FCoE/FC gateway:

- FCoE/FC gateway functionality is only supported on an OmniSwitch 6900 equipped with an OS-XNI-U12E module with SFP-FC-SR transceiver. This module provides the physical Fibre Channel interfaces that connect the OmniSwitch to native FC switches or nodes.

- FIP snooping is a required component of the OmniSwitch FCoE/FC gateway functionality. An FCoE VLAN and one or more FCoE ports are used to configure the FCoE/FC gateway fabric. For more information about how to configure these items, see Chapter 6, "Configuring FIP Snooping."

- The FCoE/FC gateway fabric defines the ports on which FC traffic is converted to FCoE traffic and FCoE traffic is converted to FC traffic. The fabric itself is defined through the association of the FC ports to an OmniSwitch VSAN and the association of FCoE ports to an FCoE VLAN. The FCoE VLAN is then mapped to the OmniSwitch VSAN to define the traffic path for converted FC or FCoE frames.
  – The FCoE/FC gateway fabric applies only to N_Port and F_Port proxy modes; the E_Port mode defines the traffic path via an E2E tunnel, which does not require a VSAN association.
  – The VSAN-to-FCoE VLAN association is a one-to-one mapping only.
  – An OmniSwitch FC port is assigned to only one VSAN at a time; however, it is possible to assign multiple FC ports to the same VSAN.

- An FCoE VLAN is a required component for an N_Port proxy operation, an F_Port Proxy operation, and a VE_Port E2E tunnel (not required for TE_Port tunnels). It is possible to use the same FCoE VLAN in all cases, but only one VE_Port tunnel per VLAN is allowed.

- An OmniSwitch VSAN only applies to the local switch configuration as a means to identify a local gateway fabric. There is no correlation between an OmniSwitch VSAN and VSANs created within a native FC SAN.

- A default VSAN 1 is provided on the switch and all FC ports are automatically assigned to that VSAN. Configuring additional VSANs is allowed to define additional gateway fabrics.

- An OmniSwitch VSAN is a required component for an N_Port proxy and F_Port proxy configuration, but a VSAN is not required to configure an E_Port proxy tunnel endpoint.

Configuring FC Ports

Configuring OS-XNI-U12E module ports with an SFP-FC-SR transceiver to ensure lossless gateway connectivity with FC SAN devices requires the following configuration tasks:

1 Assigning a Data Center Bridging (DCB) profile to each FC port.

2 Setting the SFP-FC-SR transceiver port type to Fibre Channel.

3 Configuring the Fibre Channel port mode to determine the FCoE/FC gateway operation provided on the port.

Note that the operational mode is configured on a per-port basis. This allows the OmniSwitch to provide multiple N_Port proxy, F_Port proxy, and E_Port proxy operations on the same switch.

Assigning a DCB Profile for FC Ports

Use the following steps to assign a DCB profile to each FC port on the switch:

1 Create the DCB profile using the **qos qsp dcb import** command with the **802.3x-pause** option. For example, the following command creates custom profile DCB 20, based on predefined profile 8:

```
-> qos qsp dcb 20 import qsp dcb 8 802.3x-pause
```

2 After creating the DCB profile, set the Priority Flow Control (PFC) willing bit to "no" on each FC port. For example:

```
-> qos qsi port 2/6 dcb dcbx pfc willing no
```

3 Disable PFC TLVs on each FC port. For example:

```
-> qos qsi port 2/6 dcb dcbx pfc tlv disable
```

4 Apply the custom DCB profile to each FC port. For example:

```
-> qos qsi port 2/6 qsp dcb 20
```

See Chapter 2, "Configuring Data Center Bridging," for more information.

Configuring the FC Port Type and Mode

By default, the port type for all OS-XNI-U12E module ports is set to Ethernet, even for ports equipped with the SFP-FC-SR transceiver. In addition, there is no FC operational mode set for each port. To enable FCoE/FC gateway functionality on these ports, use the **fibre-channel port mode** command. For example:

```
-> fibre-channel port 2/1 mode np
```

The **fibre-channel port mode** command sets the port type to Fibre Channel and configures the FC operational mode for the port. In the example, port 2/1 is configured as a Fibre Channel port that will provide N_Port proxy operations for the gateway switch.

Configuring an N_Port Proxy Operation

The NPIV capabilities of the OmniSwitch gateway allows the switch to serve as an N_Port proxy for ENode devices attempting to access FC switches over an FCoE network. The N_Port proxy functionality is operational when the following components are configured:

- FIP snooping is configured and enabled for the switch.

- An FCoE VLAN is mapped to a VSAN.

- At least one FC port that is assigned to the mapped VSAN is configured to operate as a proxy node port (NP_Port).

- At least one FCoE port is tagged with the FCoE VLAN.

The **fibre-channel port mode** command is used to configure an FC port to operate in the N_Port proxy mode. For example, the following command configures FC port 2/1 to operate as an NP_Port:

```
-> fibre-channel port 2/1 mode np
```

NP_Port 2/1 will connect to a native FC switch to provide N_Port login services for ENodes accessing the OmniSwitch FCoE/FC gateway through an FCoE network. To identify which ENode traffic is serviced on NP_Port 2/1, complete the following steps:

1 Create an OmniSwitch VSAN and map the VSAN to the FCoE VLAN on which ENode traffic is forwarded. In the following example, the **fibre-channel vsan** command is used to create VSAN 10 and the **fcoe vsan-map** command is used to map VSAN 10 to FCoE VLAN 500:

```
-> fibre-channel vsan 10
-> fcoe vsan-map vsan 10 vlan 500
```

2 Assign the NP_Port to the VSAN that is mapped to the FCoE VLAN using the **fibre-channel vsan members** command. For example, the following command assigns NP_Port 2/1 to VSAN 10:

```
-> fibre-channel vsan 10 members 2/1
```

3 Configure the appropriate FCoE port role for the switch ports that will receive the FCoE traffic using the **fcoe role** command. For example, the following command configures port 1/1 and 1/2 as FCoE mixed ports:

```
-> fcoe port 1/1-2 role mixed
```

4 Tag the FCoE ports configured in Step 3 with FCoE VLAN 500. For example:

```
-> vlan 500 members port 1/1-2 tagged
```

To display the N_Port proxy configuration on the OmniSwitch:

- Use the **show fibre-channel vsan** command and the **show fibre-channel vsan members** command to display the VSAN configuration and the FC ports associated with the configured VSANs.

- Use the **show fibre-channel port** command to display the operational mode and active state for each FC port. An NP_Port should be in an up state to trigger the N_Port proxy mode service.

- Use the **show fcoe vsan-map** command to determine which FCoE VLANs are mapped to which OmniSwitch VSANs.

For more information about the N_Port proxy feature, see

Configuring N_Port Proxy Load Balancing

When two NP_Ports are assigned to the same VSAN, the OmniSwitch will load balance ENode FLOGI requests on the two NP_Ports. By default, the dynamic load balancing algorithm is used. When this method is applied, the switch selects the NP_Port with the least number of logins. If all the NP_Ports in the VSAN have the same number of logins, then a round-robin selection method is used.

There are two other methods supported: dynamic re-ordering and ENode-based selection. To change the load balancing method, use the **fibre-channel npiv-proxy load-balance** command. For example,

```
-> fibre-channel npiv-proxy load-balance dynamic-reorder
-> fibre-channel npiv-proxy load-balance enode-based
```

The load balancing method is applied globally to all switch NP_Ports. To change the load balancing method applied back to dynamic, use the **default** parameter with the **fibre-channel npiv-proxy load-balance** command. For example,

```
-> fibre-channel npiv-proxy load-balance default
```

To display the load balancing information for the OmniSwitch:

- Use the **show fibre-channel** command to determine the active load balancing method for the switch.

- Use the **show fibre-channel npiv-proxy load-balance** command with the **session-count** option to display the number of sessions on each FC port.

For more information, see "N_Port Proxy Load Balancing" on page 7-10.

Statically Assigning FCoE Ports to NP Ports

To override the active load balancing method, it is possible to configure a static mapping between an FCoE port and an NP_Port. When this is done, the switch will only process ENode FLOGI requests received on the specified FCoE port by sending the related FDISC messages only on the specified NP_Port. No load balancing is applied to the statically mapped NP_Port.

To configure this type of static port mapping, use the **fibre-channel npiv-proxy load-balance static** command. For example:

```
-> fibre-channel npiv-proxy load-balance static port 1/1 fc-port 2/1
```

In this example, FCoE port 1/1 is mapped to FC port 2/1, which is configured to operate as an NP_Port. All ENode login requests received on port 1/1 are processed and directly forwarded to NP_Port 2/1.

To remove a static port mapping, use the **no** form of the **fibre-channel npiv-proxy load-balance static** command. For example,

```
-> no fibre-channel npiv-proxy load-balance static port 1/1 fc-port 2/1
```

Use the **show fibre-channel npiv-proxy load-balance** command with the **static** option to display the static FCoE port mapping configuration for the switch.

For more information, see "Statically Assigning FCoE Ports to NP Ports" on page 7-23.

Configuring an F_Port Proxy Operation

The OmniSwitch F_Port proxy functionality allows the switch to provide an F_Port fabric for FC nodes attempting to access FC switches over an FCOE network. The F_Port proxy functionality is operational when the following components are configured:

- FIP snooping is configured and enabled for the switch.

- An FCoE VLAN is mapped to a VSAN.

- At least one FC port that is assigned to the mapped VSAN is configured to operate as a fabric port (F_Port).

- At least one FCoE port (configured as an FCF only, mixed, or trusted FCoE port) tagged with the FCoE VLAN.

The **fibre-channel port mode** command is used to configure an FC port to operate in the F_Port proxy mode. For example, the following command configures FC port 2/2 to operate as an F_Port proxy:

```
-> fibre-channel port 2/2 mode f
```

To identify which FC node traffic is serviced on F_Port 2/2, complete the following steps:

1 Create an OmniSwitch VSAN and map the VSAN to the FCoE VLAN on which converted FC node traffic will be forwarded through the FCoE network. In the following example, the **fibre-channel vsan** command is used to create VSAN 10 and the **fcoe vsan-map** command is used to map VSAN 10 to FCoE VLAN 500:

```
-> fibre-channel vsan 10
-> fcoe vsan-map vsan 10 vlan 500
```

2 Assign the F_Port to the VSAN that is mapped to the FCoE VLAN using the **fibre-channel vsan members** command. For example, the following command assigns F_Port 2/2 to VSAN 10:

```
-> fibre-channel vsan 10 members 2/2
```

3 Configure at least one port as an FCF only, mixed, or trusted FCoE port using the **fcoe role** command. For example, the following command configures port 1/1 as an FCoE mixed port:

```
-> fcoe port 1/1 role mixed
```

4 Tag the FCoE ports that will forward the converted FC traffic to FCoE VLAN 500. Make sure that at least one of the FCoE ports tagged is configured as an FCF only, mixed, or trusted FCoE port. In the following example, the **vlan members tagged** command is used to tag FCoE ports 1/1 and 1/2 to FCoE VLAN 500:

```
-> vlan 500 members port 1/1-2 tagged
```

To display the F_Port proxy configuration on the OmniSwitch:

- Use the **show fibre-channel vsan** command and the **show fibre-channel vsan members** command to display the VSAN configuration and the FC ports associated with the configured VSANs.

- Use the **show fibre-channel port** command to display the operational mode for each FC port.

- Use the **show fcoe vsan-map** command to determine which FCoE VLANs are mapped to which OmniSwitch VSANs.

For more information about F_Port proxy functionality, see "Using the F_Port Proxy Mode" on page 7-11.

Configuring an E_Port Proxy Operation

The OmniSwitch gateway provides an E_Port to E_Port (E2E) tunneling function that emulates a point-to-point FC link between E_Ports on native FC switches. This E2E tunneling capability allows the OmniSwitch to serve as an E_Port proxy through which FC switches can set up inter-switch links between FC fabrics over an FCoE network.

The E_Port proxy functionality is activated when FIP snooping is configured and enabled and a tunnel endpoint is defined on each OmniSwitch gateway that will provide this functionality. The tunnel endpoints determine if the tunnel is a VE_Port tunnel that is established through an FCoE network or an internal tunnel that is established between two TE_Ports on the same switch or virtual chassis.

In both cases, configuring an OmniSwitch FC port as a tunnel expansion (TE_Port) is required. An FC TE_Port connects directly to an E_Port on an FC switch. The **fibre-channel port mode** command is used to configure an FC port to operate as a TE_Port. For example, the following command configures FC port 2/3 to operate as a TE_Port:

```
-> fibre-channel port 2/3 mode te
```

Creating a VE_Port Tunnel

To create an E2E tunnel that will traverse an FCoE network, configure a tunnel endpoint on each gateway switch that will define the tunnel path. A VE_Port tunnel endpoint consists of an FC TE_Port and an FCoE VE_Port associated with the same FCoE VLAN. The FCoE VLAN is used to tunnel traffic from the TE_Port through the FCoE network via the FCoE VE ports.

The **fcoe role** command is used to configure an FCoE port (10G Ethernet DCB port) as a VE_Port. For example, the following command designates port 1/1 as a VE_Port:

```
-> fcoe port 1/1 role ve
```

The FCoE port role is used by FIP snooping to define ACLs that are applied to the VE_Port to secure tunnel traffic through the FCoE network.

After the FCoE port is configured as a VE_Port, the port is then tagged with the designated FCoE VLAN for the tunnel. For example, the following **vlan members tagged** command tags FCoE port 1/1 with FCoE VLAN 500.

```
-> vlan 500 members port 1/1 tagged
```

Once the VE_Port is tagged with the tunnel FCoE VLAN, the FC TE_Port is then assigned to the same FCoE VLAN using the **fcoe e-tunnel** command. This command also assigns a tunnel ID for this endpoint. For example, the following command assigns TE_Port 2/3 to FCoE VLAN 500 (the same VLAN to which VE_Port 1/1 was assigned):

```
-> fcoe e-tunnel 10 fc-port1 2/3 vlan 500
```

The E2E tunnel endpoint is now configured. When the other endpoint is configured with a different tunnel ID and all associated ports are up and running, the tunnel session is established between the two endpoints via exchange link parameters (ELP). The ELP are transmitted through the FCoE network between the FC switches connected to the OmniSwitch TE_Ports associated with the same tunnel ID.

Creating a TE_Port Tunnel

To create an E2E tunnel that will directly connect two FC fabrics through the same switch or a virtual chassis configuration, use the **fcoe e-tunnel** command to create the tunnel endpoint with two TE_Ports. For example, the following command creates a tunnel between TE_Port 2/4 and TE_Port 2/5 on the same switch:

```
-> fcoe e-tunnel 20 fc-port1 2/4 fc-port2 2/5
```

In this example, the **fc-port2** parameter is used instead of the **vlan** parameter. This defines the tunnel as an internal TE_Port tunnel, instead of a VE_Port tunnel. A TE_Port tunnel does not traverse the FCoE network, so there is no need to associate the TE_Port with an FCoE VLAN. Each TE_Port is the endpoint for this type of tunnel.

To display the E2E tunnel configuration for the switch, use the **show fcoe e-tunnel** command.

For more information about the E2E tunneling feature, see "Using the E_Port Proxy Mode" on page 7-14.

FCoE/FC Gateway Configuration Examples

Example 1: Multiple Gateway Operations on the Same Fabric

The following diagram shows a sample network topology in which two OmniSwitches are operating as FCoE/FC gateway switches to secure and forward the following traffic flows:

- FCoE traffic between FCoE endpoints (ENode and FC switch)

- FC traffic between FC endpoints (FC node and FC switch)

- Inter-switch link trunking between E2E tunnel endpoints (FC switch to FC switch).

Figure 6: Sample FCoE/FC Gateway Topology

To the ENodes, FC nodes, and FC switch in this sample topology, the underlying Ethernet links through the FCoE network path appear as a direct point-to-point connection that is expected in a native FC SAN.

- The FIP snooping, E_Port proxy, and E_Port proxy operations are enabled on the OS6900-Gateway A.

- The FIP snooping, F_Port proxy, and E_Port proxy operations are enabled on the OS6900-Gateway B.

- Port 1/1 on Gateway A receives all traffic (FIP snooping, F_Port, E_Port) from Gateway B through FCoE VLAN 200. As a result, FCoE port 1/1 is configured with the FCoE port role set to mixed so that FIP snooping is applied to all traffic types.

- Port 1/1 on Gateway B also receives all traffic (FIP snooping, NP_Port, E_Port) from Gateway A through FCoE VLAN 200. As a result, FCoE port 1/1 is configured with the FCoE port role set to mixed so that FIP snooping is applied to all traffic types.

- FCoE VLAN 200 is configured through the FCoE network (manually on each OS6900 gateway switch and, if necessary, manually on the ENodes).

The following configuration information provides an example of how the FCoE/FC gateway functionality is configured for each OmniSwitch in the sample FCoE/FC gateway topology.

OS6900-Gateway A

FIP snooping is a required component for setting up an OmniSwitch FCoE/FC gateway. The following sample commands configure FIP snooping and lossless Ethernet on OS6900-Gateway A:

```
-> fcoe vlan 200
-> fcoe port 1/1 role mixed
-> vlan 200 members port 1/1 tagged
-> qos port 1/1 trusted default classification 802.1p
-> qos qsp dcb FCOE-1 import qsp dcb 7 fcoe
-> qos qsi port 1/1 qsp dcb FCOE-1
-> lldp port 1/1 tlv application enable
-> lldp port 1/1 tlv application fcoe priority 3
-> fcoe fip-snooping admin-state enable
```

By default, VSAN 1 already exists in the switch configuration and all OmniSwitch FC ports are assigned to VSAN 1. As a result, the only VSAN configuration required for this example is to map VSAN 1 to FCoE VLAN 200. For informational purposes, the commands to create the VSAN and assign the NP_Ports are included in the following sample commands used to configure N_Port proxy functionality on OS6900-Gateway A:
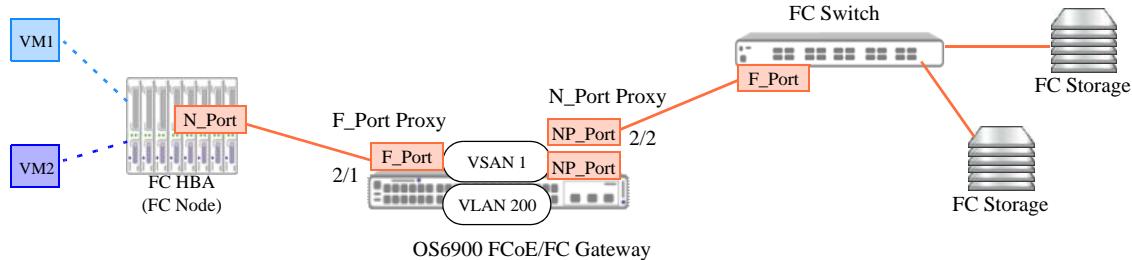
```
-> fibre-channel vsan 1
-> fcoe vsan-map vsan 1 vlan 200
-> fibre-channel fc-port 2/2 mode np
-> fibre-channel fc-port 2/3 mode np
-> fibre-channel vsan 1 members 2/2-3
```

The following sample commands configure an E_Port proxy (E2E tunnel) endpoint on OS6900-Gateway A. Note that a VSAN mapping to an FCoE VLAN is not required for E2E tunnel traffic. Instead, an FC port configured to operate as a tunnel E_Port is directly associated with FCoE VLAN 200 to carry tunnel traffic.

```
-> fibre-channel fc-port 2/1 mode te
-> fcoe e-tunnel 1
-> fcoe e-tunnel 1 fc-port1 2/1 vlan 200
```

OS6900-Gateway B

The following sample commands configure FIP snooping and lossless Ethernet on the OS6900-Gateway B switch:

```
-> fcoe vlan 200
-> fcoe port 1/1 role mixed
-> fcoe port 1/2 role edge
-> vlan 200 members port 1/1-2 tagged
-> qos port 1/1 trusted default classification 802.1p
-> qos qsp dcb FCOE-1 import qsp dcb 7 fcoe
-> qos qsi port 1/1-2 qsp dcb FCOE-1
-> lldp port 1/1-2 tlv application enable
-> lldp port 1/1-2 tlv application fcoe priority 3
-> fcoe fip-snooping admin-state enable
```

The following sample commands configure F_Port proxy functionality on the OS6900-Gateway B switch:

```
-> fibre-channel vsan 1
-> fcoe vsan-map vsan 1 vlan 200
-> fibre-channel fc-port 2/1 mode f
-> fibre-channel vsan 1 members port 2/1
```

The following sample commands configure an E_Port proxy (E2E tunnel) endpoint on the OS6900-Gateway B switch. Note that the tunnel ID used in this sample is the same as the tunnel ID used on the OS6900-Gateway A switch. This identifies both endpoints as members of the same tunnel.

```
-> fibre-channel fc-port 2/2 mode te
-> fcoe e-tunnel 1
-> fcoe e-tunnel 1 fc-port1 2/2 vlan 200
```

Example 2: OmniSwitch Gateway with NPIV Host

The following diagram shows a sample network topology in which an OmniSwitch FCoE/FC gateway provides N_Port proxy and F_Port proxy services on the same switch.

Figure 7: OmniSwitch FCoE/FC Gateway with NPIV Host

In this example, the OmniSwitch gateway emulates both an FC switch and an FCoE forwarder to allow the FC NPIV server and associated VMs to connect and access the targeted FC storage in the FC SAN.

- The F_Port proxy operation is enabled on port 2/1, which connects directly to an FC HBA port on the server. This allows the FC HBA to communicate over Ethernet (OmniSwitch gateway) with the FC switch fabric.

- The N_Port proxy operation is enabled on port 2/2, which connects directly to an F_Port on the native FC switch. This allows the OmniSwitch to also serve as an FCoE forwarder for the FC HBA sessions.

- Port 2/1 and port 2/2 are assigned to VSAN 1, which is mapped to FCoE VLAN 200.

- The OmniSwitch gateway presents an F_Port to an N_Port on the HBA and an NP_Port to the F_Port on the FC switch.

- NPIV is enabled on the FC HBA port to allow the server to request a separate FCID (also called an N_Port ID) for VM1 and VM2. This is similar to how the N_Port proxy operation uses NPIV to allow the OmniSwitch to request multiple FCIDs on the same physical port.

- Once the physical FC HBA port logs into the FC switch fabric through the OmniSwitch gateway, the server then requests additional logins for each of the VMs.

- In this scenario, the OmniSwitch gateway does not have to track FC-to-FC session reachability (does not need keep-alive messages to track the sessions). However, the OmniSwitch does maintain session information for the three login sessions (one for the HBA port and one for each VM).

- Note that there is no Ethernet FCoE port configuration needed in this example. The FCoE VLAN handles the convergence of the FC traffic for ports associated with VSAN 1. So FC over Ethernet is achieved on the same gateway switch to allow a native FC node to communicate with a native FC switch fabric.

The following **show fibre-channel sessions** command output provides a sample of how the FC HBA sessions might appear on the OmniSwitch gateway (the device represented in each session is highlighted with the same color used to represent the device in the example diagram on):

```
-> show fibre-channel sessions
Total FIBRE-CHANNEL Sessions                 : 6,
Total NPIV Sessions                          : 3,
Total R-NPIV Sessions                        : 3,
Total E-TUNNEL Sessions                      : 0

 Port    Mode VSAN T-ID      WWPN                  FCID    Status   Login Type
--------+----+----+----+---------------------+--------+--------+-------------
2/1      F    1    --   10:00:E8:E7:32:94:67:7A 01:04:01 SUCCESS  FLOGI
2/1      F    1    --   10:00:E8:E7:32:94:67:7A 01:04:03 SUCCESS  FDISC
2/1      F    1    --   10:00:E8:E7:32:94:67:7A 01:04:04 SUCCESS  FDISC
2/6      NP   1    --   10:00:E8:E7:32:94:67:7D 01:04:01 SUCCESS  FLOGI
2/6      NP   1    --   10:00:E8:E7:32:94:67:7D 01:04:03 SUCCESS  FDISC
2/6      NP   1    --   10:00:E8:E7:32:94:67:7D 01:04:04 SUCCESS  FDISC
```

In this example,

- The WWPNs are those of the OmniSwitch FC ports (F_Port 2/1 and NP_Port 2/6) on which the sessions were established.

- The "FCID" column displays a unique FCID for each device per each session type on each port. Since there are two VMs on the FC node server, there are three sessions on each OmniSwitch port: one for the physical FC HBA port and one for each VM.

- The "Login" column displays "FLOGI" or "FDISC". The FLOGI session represents the initial fabric login of the physical FC HBA port. Each FDISC session represents the fabric login for each VM. When using NPIV, the NP_Port sends FDISC requests on behalf of devices accessing the fabric on the NP_Port. The FLOGI request is for the initial fabric login of the physical NP_Port.

- Sessions are shown for the same three FCIDs on each port. This is because the port 2/1 sessions represent the F_Port sessions directly from the FC HBA and the port 2/6 sessions represent the N_Port proxy sessions requested by the OmniSwitch FCoE forwarder for the same FCIDs.

- To the FC HBA and the two VMs, the connection through the OmniSwitch gateway to the FC switch fabric is transparent. Because the HBA is also using NPIV, each VM session is uniquely identified and, therefore, separately tracked.

Verifying the FCoE/FC Gateway Configuration

Displaying the OmniSwitch FIP snooping and FCoE/FC gateway configuration is helpful to verify the actual configuration and monitor FIP snooping, N_Port proxy, F_Port proxy, and E_Port proxy sessions.

To display information about the FCoE/FC gateway configuration, use the **show** commands listed in the following table:

show fibre-channel vsan	Displays the local OmniSwitch VSAN configuration. VSANs identify the OmniSwitch FC ports as members of a specific FCoE/FC gateway fabric.
show fibre-channel vsan members	Displays the FC ports associated with each VSAN.
show fibre-channel port	Displays the OmniSwitch FC port configuration, including port operating modes, for the switch. The port mode determines the FCoE/FC gateway functionality enabled on the port.
show fcoe vsan-map	Displays the VSAN mapping configuration for the switch. VSANs are mapped to FCoE VLANs as part of the local gateway fabric.
show fibre-channel sessions	Displays the N_Port proxy, F_Port proxy, and E_Port proxy sessions associated with the local switch. Use the **clear fibre-channel sessions** command to clear sessions on all FC ports.
show fibre-channel node	Displays a list of FC nodes connected to OmniSwitch FC ports.
show fcoe e-tunnel	Displays the E2E tunnel configuration for the switch.
show fibre-channel	Displays the global FCoE/FC gateway parameter values for the switch.
show fibre-channel statistics	Displays FC port statistics. Use the **clear fibre-channel statistics** command to reset the statistics information.

To display information about the FIP snooping configuration, use the **show** commands listed in the following table:

show fcoe	Displays the global FCoE FIP snooping status and configuration for the switch.
show fcoe ports	Displays the FCoE port configuration, including port roles, for the switch. The port role determines the FIP snooping ACL configuration for the port.
show fcoe sessions	Displays the FIP sessions associated with the local switch.
show fcoe enode	Displays ENode information for the FIP sessions associated with the local switch.
show fcoe fcf	Displays FCF information for the FIP sessions associated with the local switch.
show fcoe fc-map	Displays the Fibre Channel Mapped Address Prefix (FC-MAP) configuration.
show fcoe discovery-advertisement	Displays the FIP discovery advertisement message parameter values.
show fcoe statistics	Displays ENode and FCF generated session statistics. Use the **clear fcoe statistics** command to reset the statistics information.

8 Virtual Machine Classification

Server virtualization in the data center is happening now. The OmniSwitch implements two features to help automate the discovery and movement of virtual machines (VMs):

- Virtual Network Profile (vNP)—A vNP is basically a Universal Network Profile (UNP) that will classify virtual machines in the same manner as any other device connected to a UNP port. Once a virtual machine (or any other such device) is assigned to a vNP, the virtual machine traffic is bound to a VLAN, a Shortest Path Bridging (SPB) service, or a Virtual eXtensible Local Area Network (VXLAN) service as defined in the profile. In addition, any QoS policies associated with the profile are also applied to the VM traffic.

- Edge Virtual Bridging (EVB)—based on the IEEE 802.1Qbg standard, EVB defines architecture to standardized connections between hosts with virtual machines and switches. The OmniSwitch implementation provides the ability for the switch to interact with EVB servers.
  - EVB runs between an EVB bridge (OmniSwitch running EVB) and an EVB station (server running EVB) to discover VMs running on the station or detect VM movement and associate each VM with an appropriate VLAN bridging instance.
  - The Link Layer Delivery Protocol (LLDP) is used by the EVB bridge to discover the EVB station and establish links between the station and bridge.

In This Chapter

This chapter describes the functionality provided with the vNP and EVB features and how these features are used to classify virtual machines into the OmniSwitch bridging or SPB service domain. It provides information about configuring vNP and EVB through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following topics and procedures are included in this chapter:

- "Server Virtualization Overview" on page 8-2.
- "UNP Overview" on page 8-3.
- "Using EVB" on page 8-7.
- "Tracking Virtual Machines" on page 8-12.

For more information about UNP, see Chapter 28, "Configuring Access Guardian," in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

Server Virtualization Overview

Data center virtualization, as well as the virtualization of many other types of networks, facilitates the sharing and management of network resources. The virtual network is in essence a single, logical entity that is not restricted by physical boundaries. This is similar in concept to the use of virtual local area networks (VLANs) and virtual private networks (VPNs).

A key factor in the transition of the traditional data center network is the widespread adoption of server virtualization. A virtualized server is comprised of the following main components (see Figure 1):

- Guest virtual machine—the software representation of a self-contained operating system and application software instance that runs on a host server.

- A host server—the underlying physical hardware that provides computing resources for guest virtual machines.

- Hypervisor—a software program that provisions and manages the guest virtual machines for the host server. This program enables multiple guest virtual machines to run in isolation, side-by-side on the same physical server.
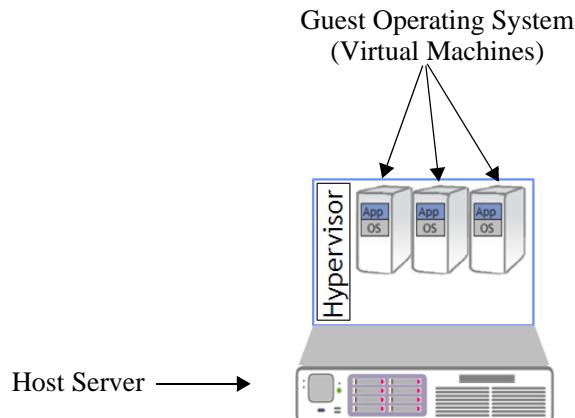
Figure 1: Virtualized Server

Server virtualization provides the following benefits:

- **Consolidation of server resources**. The ability to run multiple virtual machines on one server reduces the number of servers required to support data center applications. Using virtual machines maximizes physical server resources; the server is no longer tied to only one application or a single task.

- **Power savings**. The need for fewer servers means less power usage. In addition, existing servers can remain powered down until they are needed for virtual machine support. It is not necessary to have an underutilized server up and running all the time.

- **Hardware independence**. Virtual machines operate independently from the physical hardware on which they reside. This means that the configuration of virtual machine components (for example, RAM, CPU, network interface) can differ from the underlying components of the physical server. In addition, each virtual machine on the same physical server can run a different operating system.

- **Virtual machine mobility**. A virtual machine (VM) is basically an encapsulation of virtual hardware resources with its own operating system and applications. These resources operate independently from the underlying physical server hardware. As a result, virtual machines can easily move from one location to another without having to make underlying resource changes or disrupt live applications.

Classifying Virtual Machines

To facilitate the discovery and movement of virtual machines, the OmniSwitch provides the Universal Network Profile (UNP) feature and supports the IEEE P802.1Qbg Edge Virtual Bridging (EVB) protocol.

UNP Overview

This section provides an overview of UNP functionality, which is not restricted to supporting only data center solutions. For more information about UNP, see Chapter 28, "Configuring Access Guardian," in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

The UNP feature provides the ability to define and apply network access control to specific types of devices by grouping such devices according to specific matching profile criteria. This allows network administrators to create virtual network profiles (vNPs) *and* user network profiles from a unified framework of operation and administration.

Basically, UNP functionality is used to define profile-based VLANs or SPB service access points (SAPs) to which network devices are assigned. The profile can allow, deny, or require actions by users or machines on the network. Because membership to a VLAN or service is based on UNP profile criteria, devices assigned to the VLAN or service are not tied to a specific port or switch. This flexibility allows device mobility within the network while maintaining network security.

Implementing UNP functionality to create virtual network profiles is particularly useful in an Alcatel-Lucent Enterprise data center switching architecture. A vNP can define information such as access control rights, a VLAN or SPB service assignment, along with expected quality of service (QoS) levels for data center machines. This type of information can help virtual machines to securely bind to a VLAN bridged domain or an SPB service domain.

Profile Types

The OmniSwitch supports two separate traffic domains: VLAN and service. Traffic is classified into each domain based on the UNP profile type.

* **VLAN profile**. This type of profile creates a VLAN-port association (VPA) for device traffic that is classified into the profile. The VPA represents an association between the UNP port on which the device traffic was received and the VLAN ID specified by the profile. In other words, once classified into a specific profile, device traffic is tagged to forward on the UNP VLAN.

* **Service profile**. The OmniSwitch supports Shortest Path Bridging (SPB) services, which are based on the Provider Backbone Bridge Network (PBBN) architecture. This type of profile creates an association between device traffic that is classified into the profile and an SPB service access point (SAP).

Edge Port Types

Virtual machines bound to either a VLAN domain or a service domain can run on hosts with and without Edge Virtual Bridging (EVB) enabled. This results in three types of data center edge ports, all of which are provided through the OmniSwitch UNP and EVB features:

* UNP-enabled bridge port

* UNP-enabled service access port

* EVB-enabled bridge port

EVB service ports are not supported. See the "Using EVB" on page 8-7 for more information.

UNP also provides the ability to group UNP ports into a single logical customer domain. Once a UNP port is assigned to a specific customer domain ID, only profile classification rules associated with the same domain are applied to that port.

Using customer domains to segment network traffic ties a profile, such as a vNP, to specific UNP ports. For example, UNP ports carrying traffic for Customer A are grouped into domain 2, and profiles tailored for Customer A are also assigned to domain 2. As a result, domain 2 profiles are applied only to domain 2 UNP ports.

Classification Rules

A vNP can also define specific classification rules that are used to assign virtual machines to a profile. The following classification rules (shown in the order of precedence) are used to classify traffic received on UNP bridge and access ports:

1 MAC address + VLAN tag

2 MAC address

3 MAC address range + VLAN tag

4 MAC address range

5 IP address + VLAN tag

6 IP address

7 VLAN tag

In addition to classification rules, UNP provides the ability to trust the VLAN tag of incoming traffic. This means that virtual machines can be provisioned into the VLAN tag pre-assigned by the virtualized server hypervisor. Each UNP port can then be assigned a default UNP to classify untagged traffic when all classification mechanisms either fail or are not available.

How it Works

Dynamic assignment of devices using UNPs is achieved through port-based functionality that provides the ability to authenticate and classify device traffic. Authentication verifies the device identity and provides a UNP name. In the event authentication is not available or is unsuccessful, classification rules associated with the UNPs, as well as the UNP port configuration attributes, are applied to the traffic to determine the UNP assignment.

There is no global switch setting to invoke UNP (vNP) functionality. Instead, switch ports are configured as a UNP bridge port or a UNP access port and profiles are defined to determine the dynamic VLAN or SPB service assignment for devices connected through the UNP ports.

When UNP is enabled on a switch port or link aggregate, the following device classification process is triggered when the port receives traffic:

MAC Authentication
(RADIUS Server}

If MAC authentication fails, attempt to classify the
device using UNP rules or the packet VLAN tag.

UNP Rule or VLAN Tag
Classification

If UNP rule or VLAN tag classification fails, check
for a default profile configuration on the UNP port.

Default UNP Profile

If a default profile is not available and the
port is a UNP access port, attempt dynamic
SAP configuration, otherwise filter traffic.

Dynamic SAP
(Service Access Point)

If dynamic SAP configuration
fails, filter traffic on the port.

Figure 1: UNP Classification Overview

Virtual Network Profile Example

The following simplified illustration shows an example of vNP functionality in a virtualized network:
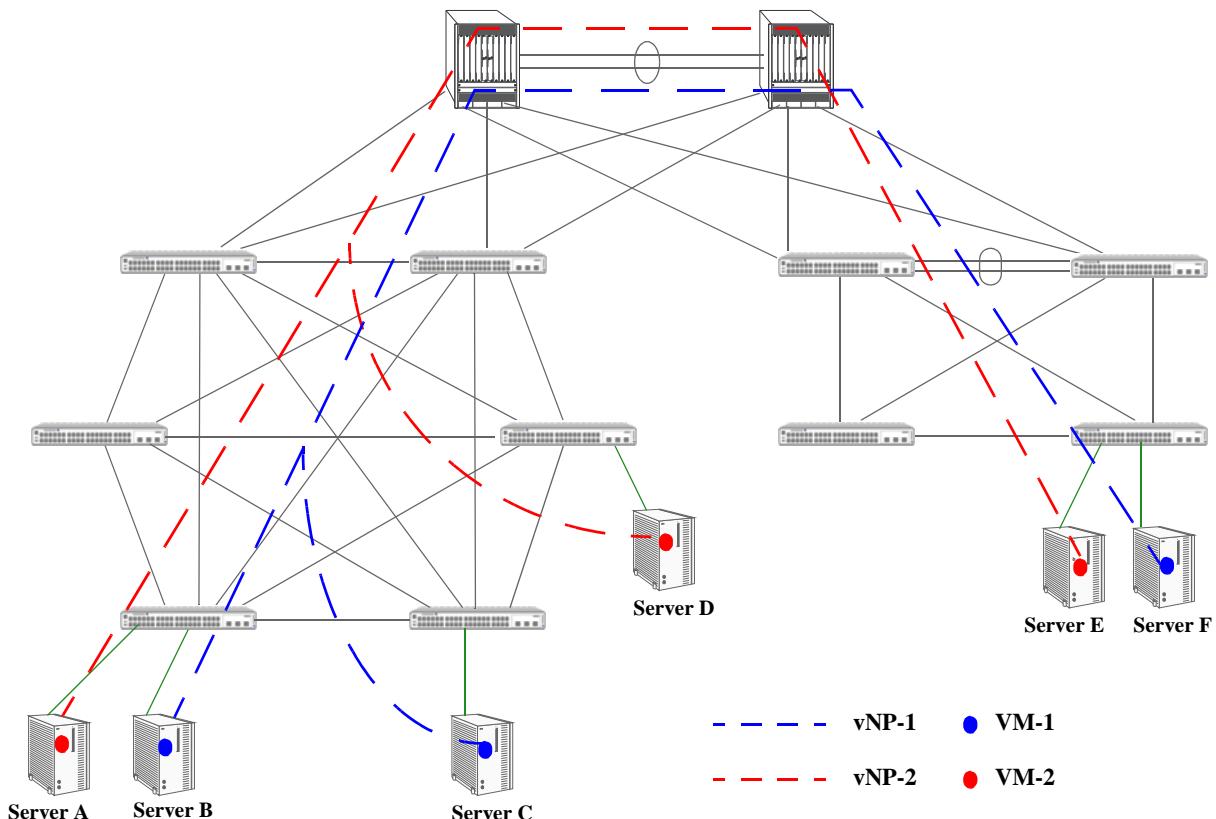
Figure 2: Virtual Network Profile–VM Mobility

In this example,

- On each switch that is connected to a server, virtual network profiles vNP-1 and vNP-2 are configured to classify and forward traffic for virtual machines VM-1 and VM-2.

- The virtual network profiles specify either a VLAN ID or service ID that will carry virtual machine traffic through the data center mesh to other servers (either within the same mesh or through the cloud to another mesh) and if any QoS policies are applied to virtual machine traffic.

- Each port connected to a server is enabled with UNP functionality.

- When the MAC address for VM-1 and for VM-2 is discovered on a UNP port, the switch applies the profile configuration to each MAC address to classify each virtual machine into a VLAN or SPB service. As a result,
  - VM-1 is assigned to the vNP-1 profile. The profile attributes then determine the VLAN or service assignment and if any QoS policies are applied to VM-1 traffic.
  - VM-2 is assigned to the vNP-2 profile. The profile attributes then determine the VLAN or service assignment and if any QoS policies are applied to VM-2 traffic.

- If VM-1 or VM-2 is moved to another server, as shown in this sample network, VM-1 will get assigned to vNP-1 and VM-2 will get assigned to vNP-2, thus the network access control configuration for each virtual machine is preserved whenever it moves to another server.

Using EVB

In the data center environment, data center servers run specialized hypervisor software to provision and manage multiple virtual machines (VMs) within the host server. These VMs can be dynamically created, deleted, or even moved to other host servers in the network.

Each VM may require different network connections and services. Typically, similar types of VMs are grouped together to form a VM network to control the unicast and multicast domains and to make a connection to storage area networks (SAN) or wide area networks (WAN).

The OmniSwitch implementation of Edge Virtual Bridging (EVB) helps automate the discovery of virtual machines and connects them to the appropriate network domain. EVB runs between a station (data center server) and a bridge (an OmniSwitch).

EVB defines logical components of the station (server): Virtual Station Interface (VSI), Edge Relay (ER), and Uplink Relay Port. The OmniSwitch supports a single ER per switch port. The ER can operate as a Virtual Ethernet Port Aggregator (VEPA) or Virtual Ethernet Bridge (VEB).

- VEPA (Virtual Ethernet Port Aggregator)—the hypervisor treats the NIC as a single interface connected to each VM and all outgoing traffic is sent through the NIC to an external switch (the Edge Relay function per IEEE 802.1Qbg in the host server).

- VEB (Virtual Ethernet Bridge)—the hypervisor creates a virtual Ethernet switch inside the host that can bridge data between VMs and send data out to the network as needed.

When a switch port is configured as an EVB bridging or service access port, then the EVB protocol is used between the station and bridge to discover VMs. A virtual switch interface (VSI) database file on the local switch is used to verify the discovered VM information. If there is a match, then EVB dynamically creates the necessary VLAN or SAP to carry the VM traffic.

1. Admin Creates VM
2. Admin adds the VM info to the VSI Database
3. Station and bridge exchange the EVB control packets for new VM
4. The VM info is retrieved from the VSI Database
5. EVB configures the Ports
6. VM Transmits packets
7. Packets are bridged to proper network
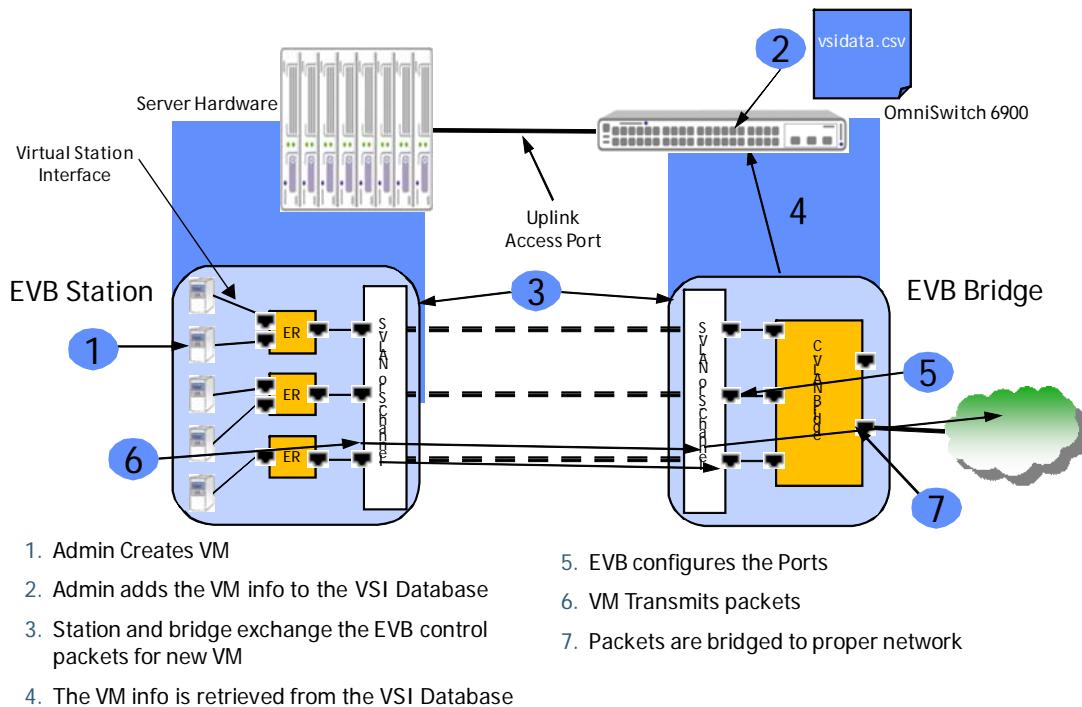
Figure 3: EVB Operational Overview

EVB ports rely on the following EVB protocols to exchange key VM VSI information:

- S-Channel Discovery and Configuration Protocol (CDCP)—Used to set up S-Channels, which are needed when a station has two or more ERs.

- EVB TLVs—Used to advertise and negotiate the EVB capabilities of a station or bridge.

- VSI Discovery and Configuration Protocol (VDP)—Assigns VSI instances to C-VLANs. Assigning multiple VLANs and MAC addresses to the same VSI instance is allowed.

- Edge Control Protocol (ECP)—Provides reliable delivery of VDP DUs between the station and the bridge.

If EVB protocol fails, then the VM connection will fail and no packets will be sent to the switch.

VSI Database

The OmniSwitch VSI database is managed and maintained by the UNP application. The EVB uses information collected during the VSI Discovery Protocol (VDP) exchange between the EVB OmniSwitch and the EVB station (server) to send a VSI database query to the UNP application. UNP compares the VSI database entries to the values in the EVB query. If there is a match, UNP provides the matching VSI entry values to EVB. In this manner, EVB confirms the VSI information discovered and gets the C-VLAN and/or priority information that is then used to classify the VM on the OmniSwitch.

In essence, the VSI database serves as a VM reservation list on the OmniSwitch. The EVB exchange between the switch and the station is done to confirm the reservation and obtain classification information for the VM. See "Creating the VSI Database File" on page 8-10 for more information.

EVB Configuration Guidelines

Only the EVB VLAN bridging mode is supported at this time. In this mode,

- The VM requests specific C-VLAN(s). Note that the VM must provide the requested C-VLANs when it sends EVB protocol packets to the switch.

- The switch dynamically creates the corresponding VLANs and propagates those VLANs to the other switches in the network using the MVRP protocol.

- After the VM is associated to the bridge using the EVB protocol, then the VM can send data traffic tagged with the requested C-VLAN.

- EVB does not support the use of GroupID and S-channel in the VLAN bridging mode.

UNP and EVB are mutually exclusive on the same port; only UNP or EVB functionality is allowed on a specific port or link aggregate at any given time.

- If a port is configured as an EVB bridging port, then the EVB protocol is used between the station and bridge to discover VMs.

- If a port is configured as a UNP bridging or access port, then the regular UNP classification methods are used to assign the VM traffic to a dynamic VLAN or dynamic SAP.

- The UNP method is mainly used for stations that are not using EVB.

- The EVB method is only for interaction with EVB stations.

Configuring EVB

Configuring EVB functionality on the OmniSwitch involves the following steps:

1 Create the VSI database. The **vsidata.csv** file is a user-configured and maintained database that contains the virtual switch interface (VSI) information for VMs. This file must be in the **.csv** format and contain specific identifying information that is used to classify VMs. See "Creating the VSI Database File" on page 8-10 for more information.

2 Upload the VSI database file to the switch. The **vsidata.csv** file must reside in the **/flash/vsidata/** directory on the local switch. If this directory does not exist, then manually create it on the switch making sure to use "vsidata" as the directory name. Upload the vsidata.csv file into the /flash/vsidata/ directory.

3 Load the VSI database file into switch memory. Once the vsidata.csv file is created and uploaded tot he /flash/vsidata/ directory, use the **unp reload vsi-type-database** command to load the file contents into switch memory. For example:

```
-> unp reload vsi-type-database
```

Note that any time a new VSI file is uploaded or the existing file is changed, a file reload into switch memory is required to apply the updated file.

4 Configure a switch port or link aggregate as an EVB port. There are two methods for configuring EVB ports: dynamic or manual configuration of EVB on the port.

The dynamic option is a global option that applies to all switch ports. When enabled, any port that receives S-Channel Discovery and Configuration Protocol (CDCP) or EVB TLV (type, length, value) control packets, is automatically configured as an EVB port. The manual option applies EVB directly to a specific port.

To enable dynamic configuration of a port as an EVB bridge or access port, use the **evb port auto enable type** command with the **vlan-bridging** or **service-access** option. For example:

```
-> evb port auto enable type vlan-bridging
-> evb port auto enable type service-access
```

To manually configure a port as an EVB bridge or access port, use the **evb port type** command with the **vlan-bridging** or **service-access** option. For example:

```
-> evb port 1/1 type vlan-bridging
-> evb port 1/2 type service-access
```

5 *Optional*. Modify the following EVB parameter values that are set by default at the time the EVB session is created. If necessary, use the specified commands to change the default values.

Parameter Description	Command	Default
Use local EVB TLV settings instead of exchanging local and remote EVB TLV settings to determine the operational configuration.	**evb evb-lldp manual**	Disabled
The amount of time to wait for an EVB Edge Control Protocol (ECP) acknowledgment (ACK).	**evb ecp default-ack-timer**	14 (approximately 163.84 milliseconds)

Parameter Description	Command	Default
The maximum number of times to retransmit ECP packets when no ACK is received. Retransmit is triggered after the ACK timer expires.	**evb ecp default-max-retry**	3
The VSI Discovery Protocol (VDP) resource timeout	**evb vdp default-resource-wait-delay**	20 (approximately 10.48576 seconds)
The VDP keep alive timeout	**evb vdp default-keep-alive-timeout**	20 (approximately 10.48576 seconds)

Creating the VSI Database File

The network administrator manually defines a VSI database entry in the **vsidata.csv** file for each VSI instance and then copies that file to the **/flash/vsidata** directory on each OmniSwitch bridge that is connected to EVB stations. Each database entry consists of the following values:

- VSI Manager ID
- VSI Type
- VSI Type Version
- VSI Instance ID
- Return Entry Type
- VLAN
- Priority
- MAC
- Group ID

When creating the **vsidat.csv** file and VSI entries, use the following required guidelines:

- The first row of the VSI database file must contain the following column names in the following order:

- VSI Manager ID, VSI Instance ID and MAC address should be hexadecimal only. All other fields should be decimal.

- Duplicate entries are not allowed.

- Number of return entries are up to sixteen.

- Wild cards are allowed on VSI Instance ID and VSI Type fields only

- For VSI Manager ID and VSI Instance ID, the delimiter should be space. For MAC, the delimiter should be colon. For all other entries, the delimiter should be comma.

- The data file name must be **vsidata.csv** and located in **/flash/vsidata/** directory.

The following shows the contents and layout of a sample **vsidata.csv** file:

	A	B	C	D	E	F	G	H	I
1	VSI Manager ID	VSI Type	VSI Type Version	VSI Instance ID	Type	VLAN	Priority	MAC	Group ID
2	20 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	1	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	1	100	1		
3	30 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	12	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	2	200	2	aa:bb:cc:dd:ee:ff	
4	10 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	13	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	1	300,500,600,700	1,4,5,7		4
5	11 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	14	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	3	400	3		5,110,120,130
6	12 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	15	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	1	100,200,300	1,2,3		
7	13 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	16	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	2	19,20,21	30,40,50	aa:cc:bb:ee:dd:ff	
8	14 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	11	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	4	11,12,13,14,15,16,17,18	1,2,3,4,5,6,7,8	22:33:44:55:66:77	101,102,103
9	14 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01	*	0	12 23 34 45 56 67 78 90 01 12 23 34 45 56 67 78	1	199	2		
10	14 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	*	9	*	1	211,222	4,5		
11	14 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	*	7	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	1	777,111,222	3,4,5		
12	20 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	20	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	1	100	1		
13	30 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	21	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	2	200	2	aa:bb:cc:dd:ee:ff	
14	10 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	22	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	1	300,500,600,700	1,4,5,7		4
15	11 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	23	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	3	400	3		5,110,120,130
16	12 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	24	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	1	100,200,300	1,2,3		
17	13 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	25	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	2	19,20,21	30,40,50	aa:cc:bb:ee:dd:ff	
18	14 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	26	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	4	11,12,13,14,15,16,17,18	1,2,3,4,5,6,7,8	22:33:44:55:66:77	101,102,103
19	20 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	27	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	1	100	1		
20	30 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	28	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	2	200	2	aa:bb:cc:dd:ee:ff	
21	10 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	29	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	1	300,500,600,700	1,4,5,7		4
22	11 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	30	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	3	400	3		5,110,120,130
23	12 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	31	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	1	100,200,300	1,2,3		
24	13 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	32	1	C1 11 20 DE 07 2B 11 E1 85 A1 00 10 94 00 00 03	2	19,20,21	30,40,50	aa:cc:bb:ee:dd:ff	

UNP parses the VSI manager ID with a three-field key (VSI Type; VSI Type Version; VSI Instance ID).

On an EVB bridged port, only the VLAN ID from the database is matched with the VLAN tag pre-assigned by the hypervisor with UNP trust-tag functionality. The VM is classified into the matching profile VLAN and associated QoS policy list.

- If UNP dynamic VLAN configuration is enabled, the switch will dynamically create the VLAN if necessary.

- If UNP dynamic profile configuration is enable, the switch will dynamically create the vNP profile if necessary.

On an EVB service access port, the VLAN ID and Group ID fields are used to determine the matching profile. UNP will request the service manager to dynamically create the SPB service on this switch if:

- The expected service does not exist;

- The system level dynamic-service-configuration is enabled; and

- The profile has the information required to create the service.

See the Chapter 3, "Configuring Shortest Path Bridging," and Chapter 28, "Configuring Access Guardian," for more information about SPB services and vNP profiles.

Tracking Virtual Machines

Virtual hypervisor managers can perform virtual machine (VM) movement without intervention. When this type of movement occurs, the hypervisor manager indicates to which hypervisor (virtual server) the virtual machine was moved. However, it is not easy to tell which OmniSwitch port in the data center mesh the virtual machine is now using to access the network.

Alcatel-Lucent Enterprise network management tools provide management and visibility for virtual machines in the mesh. The suite of management tools includes:

- OmniVista 2500 Virtual Machine Management (VMM) for management of VM mobility and integration with standard hypervisors.

- OmniVista 2500 NMS for mesh provisioning and management.

OmniVista 2500 provides a single pane of management for VMs across the network, which includes VM visibility and their point of association to the network. This gives network administrators a dashboard to determine and track the following information:

- Virtual machine locations.

- Virtual machine types.

- The switch ports to which the virtual machines are connected.

- The duration of the connections.

- Which virtual network profile (vNP) the virtual machine is using.

VMM is an embedded application within OmniVista that consists of the following components:

- **vCenter integration**: VM discovery, VM polling and event listener service will listen to preference changes and event notifications.

- **vNP configuration**: Profiles of UNP configuration that specify the configuration associated with each VM VLAN. The profiles can be assigned to switches, as and when needed.

- **VLAN Notification**: Displays a list of VM instances where the network is missing some configuration to effectively handle VMs attached. This allows the administrator to be proactive in identifying and correcting any necessary network configuration.

- **OmniVista Locator**: Provides the ability to search for specific VMs using various search criteria.

A Software License and Copyright Statements

This appendix contains ALE USA, Inc. and third-party software vendor license and copyright statements.

ALE USA, Inc. License Agreement

ALE USA, INC. SOFTWARE LICENSE AGREEMENT

IMPORTANT. Please read the terms and conditions of this license agreement carefully before opening this package.

By opening this package, you accept and agree to the terms of this license agreement. If you are not willing to be bound by the terms of this license agreement, do not open this package. Please promptly return the product and any materials in unopened form to the place where you obtained it for a full refund.

1. **License Grant.** This is a license, not a sales agreement, between you (the "Licensee") and ALE USA, Inc. ALE USA, Inc. hereby grants to Licensee, and Licensee accepts, a non-exclusive license to use program media and computer software contained therein (the "Licensed Files") and the accompanying user documentation (collectively the "Licensed Materials"), only as authorized in this License Agreement. Licensee, subject to the terms of this License Agreement, may use one copy of the Licensed Files on the Licensee's system. Licensee agrees not to assign, sublicense, transfer, pledge, lease, rent, or share their rights under this License Agreement. Licensee may retain the program media for backup purposes with retention of the copyright and other proprietary notices. Except as authorized under this paragraph, no copies of the Licensed Materials or any portions thereof may be made by Licensee and Licensee shall not modify, decompile, disassemble, reverse engineer, or otherwise attempt to derive the Source Code. Licensee is also advised that ALE USA, Inc. products contain embedded software known as firmware which resides in silicon. Licensee may not copy the firmware or transfer the firmware to another medium.

2. **ALE USA, Inc.'s Rights.** Licensee acknowledges and agrees that the Licensed Materials are the sole property of ALE USA, Inc. and its licensors (herein "its licensors"), protected by U.S. copyright law, trademark law, and are licensed on a right to use basis. Licensee further acknowledges and agrees that all rights, title, and interest in and to the Licensed Materials are and shall remain with ALE USA, Inc. and its licensors and that no such right, license, or interest shall be asserted with respect to such copyrights and trademarks. This License Agreement does not convey to Licensee an interest in or to the Licensed Materials, but only a limited right to use revocable in accordance with the terms of this License Agreement.

3. **Confidentiality.** ALE USA, Inc. considers the Licensed Files to contain valuable trade secrets of ALE USA, Inc., the unauthorized disclosure of which could cause irreparable harm to ALE USA, Inc. Except as expressly set forth herein, Licensee agrees to use reasonable efforts not to disclose the Licensed Files to any third party and not to use the Licensed Files other than for the purpose authorized by this License Agreement. This confidentiality obligation shall continue after any termination of this License Agreement.

4. **Indemnity.** Licensee agrees to indemnify, defend and hold ALE USA, Inc. harmless from any claim, lawsuit, legal proceeding, settlement or judgment (including without limitation ALE USA, Inc.'s reasonable United States and local attorneys' and expert witnesses' fees and costs) arising out of or in connection with the unauthorized copying, marketing, performance or distribution of the Licensed Files.

5. **Limited Warranty.** ALE USA, Inc. warrants, for Licensee's benefit alone, that the program media shall, for a period of ninety (90) days from the date of commencement of this License Agreement (referred to as the Warranty Period), be free from defects in material and workmanship. ALE USA, Inc. further warrants, for Licensee benefit alone, that during the Warranty Period the Licensed Files shall operate substantially in accordance with the functional specifications in the User Guide. If during the Warranty Period, a defect in the Licensed Files appears, Licensee may return the Licensed Files to ALE USA, Inc. for either replacement or, if so elected by ALE USA, Inc., refund of amounts paid by Licensee under this License Agreement. EXCEPT FOR THE WARRANTIES SET FORTH ABOVE, THE LICENSED MATERIALS ARE LICENSED "AS IS" AND ALE USA, INC. AND ITS LICENSORS DISCLAIM ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING (WITHOUT LIMITATION) ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO LICENSEE. THIS WARRANTY GIVES THE LICENSEE SPECIFIC LEGAL RIGHTS. LICENSEE MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

6. **Limitation of Liability.** ALE USA, Inc.'s cumulative liability to Licensee or any other party for any loss or damages resulting from any claims, demands, or actions arising out of or relating to this License Agreement shall not exceed the license fee paid to ALE USA, Inc. for the Licensed Materials. IN NO EVENT SHALL ALE USA, INC. BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, OR EXEMPLARY DAMAGES OR LOST PROFITS, EVEN IF ALE USA, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION TO INCIDENTAL OR CONSEQUENTIAL DAMAGES MAY NOT APPLY TO LICENSEE.

7. **Export Control.** This product is subject to the jurisdiction of the United States. Licensee may not export or reexport the Licensed Files, without complying with all United States export laws and regulations, including but not limited to (i) obtaining prior authorization from the U.S. Department of Commerce if a validated export license is required, and (ii) obtaining "written assurances" from licensees, if required.

8. **Support and Maintenance.** Except as may be provided in a separate agreement between ALE USA, Inc. and Licensee, if any, ALE USA, Inc. is under no obligation to maintain or support the copies of the Licensed Files made and distributed hereunder and ALE USA, Inc. has no obligation to furnish Licensee with any further assistance, documentation or information of any nature or kind.

9. **Term.** This License Agreement is effective upon Licensee opening this package and shall continue until terminated. Licensee may terminate this License Agreement at any time by returning the Licensed Materials and all copies thereof and extracts therefrom to ALE USA, Inc. and certifying to ALE USA, Inc. in writing that all Licensed Materials and all copies thereof and extracts therefrom have been returned or erased by the memory of Licensee's computer or made non-readable. ALE USA, Inc. may terminate this License Agreement upon the breach by Licensee of any term hereof. Upon such termination by

ALE USA, Inc., Licensee agrees to return to ALE USA, Inc. or destroy the Licensed Materials and all copies and portions thereof.

10. **Governing Law.** This License Agreement shall be construed and governed in accordance with the laws of the State of California.

11. **Severability.** Should any term of this License Agreement be declared void or unenforceable by any court of competent jurisdiction, such declaration shall have no effect on the remaining terms herein.

12. **No Waiver.** The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

13. **Notes to United States Government Users.** Software and documentation are provided with restricted rights. Use, duplication or disclosure by the government is subject to (i) restrictions set forth in GSA ADP Schedule Contract with ALE USA, Inc.'s reseller(s), or (ii) restrictions set forth in subparagraph (c) (1) and (2) of 48 CFR 52.227-19, as applicable.

14.**Third Party Materials.** Licensee is notified that the Licensed Files contain third party software and materials licensed to ALE USA, Inc. by certain third party licensors. Some third party licensors are third part beneficiaries to this License Agreement with full rights of enforcement. Please refer to the section entitled "Third Party Licenses and Notices" on page -4 for the third party license and notice terms.

Third Party Licenses and Notices

Legal Notices applicable to any software distributed alone or in connection with the product to which this document pertains, are contained in files within the software itself located at: **/flash/foss**.

Also, if needed, we provide all FOSS (Free and Open Source Software) source code used into this release at the following URL: https://service.esd.alcatel-lucent.com/portal/page/portal/EService/release

Index