Part No. 033564-00 Rev. A
April 2021

OmniSwitch AOS Release 8 Advanced Routing Configuration Guide

8.7R2

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 8.7R2.
The functionality described in this guide is subject to change without notice.

The Alcatel-Lucent name and logo are trademarks of Nokia used under license by ALE. To view other trademarks used by affiliated companies of ALE Holding, visit: www.al-enterprise.com/en/legal/trademarks-copyright. All other trademarks are the property of their respective owners. The information presented is subject to change without notice. Neither ALE Holding nor any of its affiliates assumes any responsibility for inaccuracies contained herein.

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: businessportal2.alcatel-lucent.com
Email: ebg_global_supportcenter@al-enterprise.com

Contents

List of Figures

About This Guide

This *OmniSwitch AOS Release 8 Advanced Routing Configuration Guide* describes how to set up and monitor advanced routing protocols for operation in a live network environment. The routing protocols described in this manual 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 on how fundamental software features are implemented in the OmniSwitch Series switches will benefit from the material in this configuration guide.

When Should I Read this Manual?

Read this guide as soon as you are ready to integrate your OmniSwitch into your network and you are ready to set up advanced routing protocols. 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. This manual will help you set up your switches to route on the network using routing protocols, such as OSPF.

What is in this Manual?

This configuration guide includes information about configuring the following features:

- Open Shortest Path First (OSPF) protocol
- Border Gateway Protocol (BGP)
- Multicast routing boundaries
- Distance Vector Multicast Routing Protocol (DVMRP)
- Protocol-Independent Multicast (PIM)—Sparse Mode, Dense Mode, and Source-Specific Multicast

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 CLI commands, consult the *OmniSwitch AOS Release 8 CLI Reference Guide.*

How is the Information Organized?

Chapters in this guide are broken down by software feature. The titles of each chapter include protocol or feature names (e.g., OSPF, PIM) with which most network professionals are familiar.

Each software feature chapter 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. Most chapters also 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 a *Quick Steps* section, which is a procedure covering the basic steps required to get a software feature up and running.

In-Depth Information. All chapters include *overview sections* on the software feature 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. Serious readers and advanced users will also find the many *application examples*, located near the end of chapters, helpful. Application examples include diagrams of real networks and then provide solutions using the CLI to configure a particular feature, or more than one feature, within the illustrated network.

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) and Data Center Bridging protocols (PFC, ETC, and DCBX).

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 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: businessportal2.alcatel-lucent.com

Phone: 1-800-995-2696

Email: ebg_global_supportcenter@al-enterprise.com

1 Configuring OSPF

Open Shortest Path First routing (OSPF) is a shortest path first (SPF), or *link state*, protocol. OSPF is an interior gateway protocol (IGP) that distributes routing information between routers in a single Autonomous System (AS). OSPF chooses the least-cost path as the best path. OSPF is suitable for complex networks with large numbers of routers since it provides faster convergence where multiple flows to a single destination can be forwarded on one or more interfaces simultaneously.

In This Chapter

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

Configuration procedures described in this chapter include:

- Loading and enabling OSPF (see page 1-14).
- Creating OSPF areas (see page 1-15).
- Creating OSPF interfaces (see page 1-18).
- Creating virtual links (see page 1-21).
- Configuring redistribution using route maps (see page 1-22).
- Converting Local Interfaces into OSPF Passive Interface Using Route Map (see page 1-28).

For information on creating and managing VLANs, see “Configuring VLANs” in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

OSPF Defaults

The following table shows the default settings of the configurable OSPF parameters:

Parameter Description	Command	Default Value/Comments
Enables OSPF.	**ip ospf admin-state**	disabled
Enables an interface.	**ip ospf interface admin-state**	disabled
Sets the overflow interval value.	**ip ospf exit-overflow-interval**	0
Assigns a limit to the number of External Link-State Database (LSDB) entries.	**ip ospf extlsdb-limit**	-1
Configures timers for Shortest Path First (SPF) calculation.	**ip ospf spf-timer**	delay: 5 hold: 10
Creates or deletes an area default metric.	**ip ospf area default-metric**	ToS: 0 Type: OSPF Cost: 1
Configures OSPF interface dead interval.	**ip ospf interface dead-interval**	40 seconds (broadcast and point-to-point) 120 seconds (NBMA and point-to-multipoint)
Configures OSPF interface hello interval.	**ip ospf interface hello-interval**	10 seconds (broadcast and point-to-point) 30 seconds (NBMA and point-to-multipoint)
Configures the OSPF interface cost.	**ip ospf interface cost**	1
Configures the OSPF poll interval.	**ip ospf interface poll-interval**	120 seconds
Configures the OSPF interface priority.	**ip ospf interface priority**	1
Configures OSPF interface retransmit interval.	**ip ospf interface retrans-interval**	5 seconds
Configures the OSPF interface transit delay.	**ip ospf interface transit-delay**	1 second
Configures the OSPF interface type.	**ip ospf interface type**	broadcast
Configures support for the graceful restart feature on an OSPF router.	**ip ospf restart-support**	disabled

OSPF Quick Steps

The followings steps are designed to show the user the necessary set of commands for setting up a router to use OSPF:

1 Create a VLAN using the **vlan** command. For example:

```
-> vlan 5
-> vlan 5 admin-state enable
```

2 Assign a router IP address and subnet mask to the VLAN using the **ip interface** command. For example:

```
-> ip interface vlan-5 vlan 5 address 120.1.4.1 mask 255.0.0.0
```

3 Assign a port to the created VLANs using the **vlan members** command. For example:

```
-> vlan 5 members port 2/1 untagged
```

Note. The port will be statically assigned to the VLAN, as a VLAN must have a physical port assigned to it in order for the router port to function. However, the router could be set up in such a way that ports are dynamically assigned to VLANs using classification rules (see the "Access Guardian" chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide*).

---

4 Assign a router ID to the router using the **ip router router-id** command. For example:

```
-> ip router router-id 1.1.1.1
```

5 Load and enable OSPF using the **ip load ospf** and the **ip ospf admin-state** commands. For example:

```
-> ip load ospf
-> ip ospf admin-state enable
```

6 Create a backbone to connect this router to others, and an area for the router's traffic, using the **ip ospf area** command. (Backbones are always labeled area 0.0.0.0.) For example:

```
-> ip ospf area 0.0.0.0
-> ip ospf area 0.0.0.1
```

7 Create an OSPF interface for each VLAN created in Step 1, using the **ip ospf interface** command. The OSPF interface should use the same interface name used for the VLAN router IP created in Step 2. For example:

```
-> ip ospf interface vlan-5
```

Note. The interface name *cannot* have spaces.

---

8 Assign the OSPF interface to the area and the backbone using the **ip ospf interface area** command. For example:

```
-> ip ospf interface vlan-5 area 0.0.0.0
```

9 Enable the OSPF interfaces using the **ip ospf interface admin-state** command. For example:

```
-> ip ospf interface vlan-5 admin-state enable
```

10 You can now display the router OSPF settings by using the **show ip ospf** command. The output generated is similar to the following:

```
-> show ip ospf
  Router Id                          = 46.46.46.46,
  OSPF Version Number                = 2,
  Admin Status                       = Enabled,
  Area Border Router ?               = No,
  AS Border Router Status            = Enabled,
  Route Tag                          = 0,
  SPF Hold  Time (in seconds)        = 10,
  SPF Delay  Time (in seconds)       = 5,
  MTU Checking                       = Disabled,
  # of Routes                        = 197,
  # of AS-External LSAs              = 0,
  # of self-originated LSAs          = 2,
  # of LSAs received                 = 102,
  External LSDB Limit                = -1,
  Exit Overflow Interval             = 0,
  # of SPF calculations done         = 2,
  # of Incr SPF calculations done    = 0,
  # of Init State Nbrs               = 0,
  # of 2-Way State Nbrs              = 0,
  # of Exchange State Nbrs           = 0,
  # of Full State Nbrs               = 2,
  # of attached areas                = 1,
  # of Active areas                  = 1,
  # of Transit areas                 = 0,
  # of attached NSSAs                = 1,
  Default Route Origination          = none,
  Default Route Metric-Type/Metric   = type2
  BFD Status                         = Disabled
```

Router ID
As set in Step 4

11 You can display OSPF area settings using the **show ip ospf area** command. For example:

```
-> show ip ospf area 0.0.0.0
Area Identifier                           = 0.0.0.0,
Admin Status                              = Enabled,
Operational Status                        = Up,
Area Type                                 = normal,
Area Summary                              = Enabled,
Time since last SPF Run                   = 00h:08m:37s,
# of Area Border Routers known            = 1,
# of AS Border Routers known              = 0,
# of Active Virtual Links                 = 0,
# of LSAs in area                         = 1,
# of SPF Calculations done                = 1,
# of Incremental SPF Calculations done    = 0,
# of Neighbors in Init State              = 0,
# of Neighbors in 2-Way State             = 0,
# of Neighbors in Exchange State          = 0,
# of Neighbors in Full State              = 0,
# of Interfaces attached                  = 1
Attached Interfaces                       = intf101,
```

Area ID
As set in Step 6

12 You can display OSPF interface settings using the **show ip ospf interface** command. For example:

```
-> show ip ospf interface vlan-5
Interface IP Name                       = vlan-3
Interface IP Address                    = 120.1.4.1,
Interface IP Mask                       = 255.0.0.0,
Domain Name                             = Vlan
Domain ID                               = 3
Admin Status                            = Enabled,
Operational Status                      = Down,
OSPF Interface State                    = Down,
Interface Type                          = Broadcast,
Area Id                                 = 0.0.0.0,
Designated Router IP Address            = 0.0.0.0,
Designated Router RouterId              = 0.0.0.0,
Backup Designated Router IP Address     = 0.0.0.0,
Backup Designated Router RouterId       = 0.0.0.0,
MTU (bytes)                             = 1500,
Metric Cost                             = 1,
Priority                                = 1,
Hello Interval (seconds)                = 10,
Transit Delay (seconds)                 = 1,
Retrans Interval (seconds)              = 5,
Dead Interval (seconds)                 = 40,
Poll Interval (seconds)                 = 120,
Link Type                               = Broadcast,
Authentication Type                     = simple,
Authentication Key                      = Set,
# of Events                             = 0,
# of Init State Neighbors               = 0,
# of 2-Way State Neighbors              = 0,
# of Exchange State Neighbors           = 0,
# of Full State Neighbors               = 0
BFD status                              = Disabled,
DR-Only Option for BFD                  = Disabled
```

Interface ID
As set in Step 2

VLAN ID
As set in Step 1

Interface Status
As set in Step 9

Area ID
As set in Step 8

OSPF Overview

Open Shortest Path First routing (OSPF) is a shortest path first (SPF), or link-state, protocol. OSPF is an interior gateway protocol (IGP) that distributes routing information between routers in a Single Autonomous System (AS). OSPF chooses the least-cost path as the best path.

Each participating router distributes its local state (i.e., the router's usable interfaces, local networks, and reachable neighbors) throughout the AS by flooding. In a link-state protocol, each router maintains a database describing the entire topology. This database is built from the collected link state advertisements of all routers. Each multi-access network that has at least two attached routers has a designated router and a backup designated router. The designated router floods a link state advertisement for the multi-access network.

When a router starts, it uses the OSPF Hello Protocol to discover neighbors. The router sends Hello packets to its neighbors, and in turn receives their Hello packets. On broadcast and point-to-point networks, the router dynamically detects its neighboring routers by sending Hello packets to a multicast address. On non-broadcast and point-to-multipoint networks, some configuration information is necessary in order to configure neighbors. On all networks (broadcast or non-broadcast), the Hello Protocol also elects a designated router for the network.

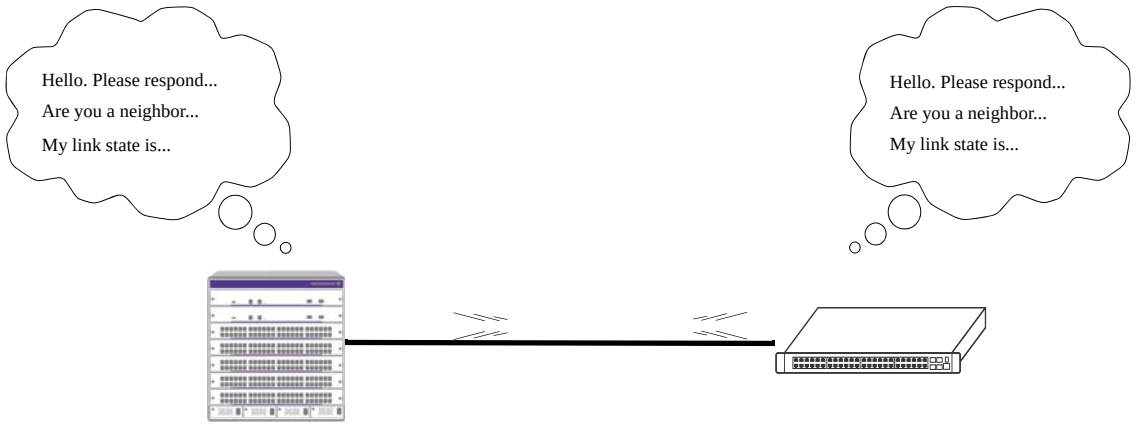


Figure 1-1 : OSPF Hello Protocol

The router will attempt to form full adjacencies with all of its newly acquired neighbors. Only some pairs, however, will be successful in forming full adjacencies. Topological databases are synchronized between pairs of fully adjacent routers.

Adjacencies control the distribution of routing protocol packets. Routing protocol packets are sent and received only on adjacencies. In particular, distribution of topological database updates proceeds along adjacencies.

Link state is also advertised when a router's state changes. A router's adjacencies are reflected in the contents of its link state advertisements. This relationship between adjacencies and link state allows the protocol to detect downed routers in a timely fashion.

Link state advertisements are flooded throughout the AS. The flooding algorithm ensures that all routers have exactly the same topological database. This database consists of the collection of link state advertisements received from each router belonging to the area. From this database each router calculates a shortest-path tree, with itself as root. This shortest-path tree in turn yields a routing table for the protocol.

OSPF Areas

OSPF allows collections of contiguous networks and hosts to be grouped together as an *area*. Each area runs a separate copy of the basic link-state routing algorithm (usually called SPF). This means that each area has its own topological database, as explained in the previous section.

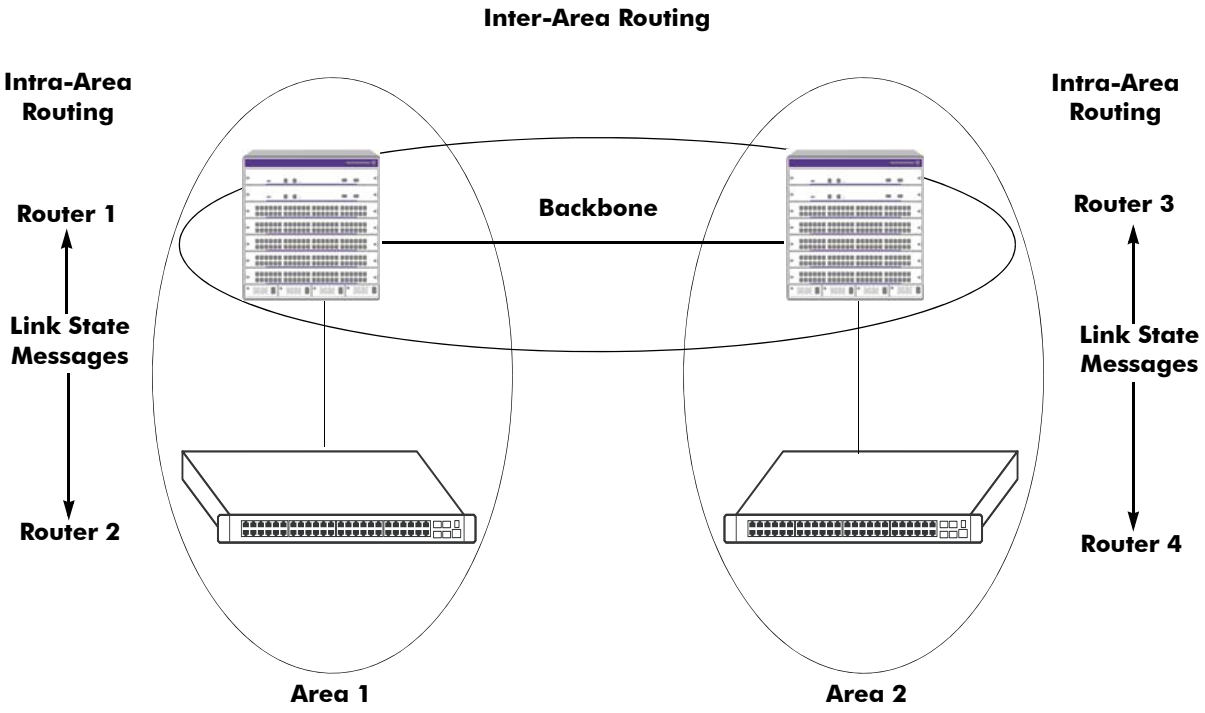


Figure 1-2 : OSPF Intra-Area and Inter-Area Routing

An area's topology is visible only to the members of the area. Conversely, routers internal to a given area know nothing of the detailed topology external to the area. This isolation of knowledge enables the protocol to reduce routing traffic by concentrating on small areas of an AS, as compared to treating the entire AS as a single link-state domain.

Areas cause routers to maintain a separate topological database for each area to which they are connected. (Routers connected to multiple areas are called *area border routers*). Two routers belonging to the same area have identical area topological databases.

Different areas communicate with each other through a *backbone*. The backbone consists of routers with contacts between multiple areas. A backbone must be contiguous (i.e., it must be linked to all areas).

The backbone is responsible for distributing routing information between areas. The backbone itself has all of the properties of an area. The topology of the backbone is invisible to each of the areas, while the backbone itself knows nothing of the topology of the areas.

All routers in an area must agree on that area's parameters. Since a separate copy of the link-state algorithm is run in each area, most configuration parameters are defined on a per-router basis. All routers belonging to an area must agree on that area's configuration. Misconfiguration will keep neighbors from forming adjacencies between themselves, and OSPF will not function.

Classification of Routers

When an AS is split into OSPF areas, the routers are further divided according to function into the following four overlapping categories:

- **Internal routers**. A router with all directly connected networks belonging to the same area. These routers run a single copy of the SPF algorithm.
- **Area border routers**. A router that attaches to multiple areas. Area border routers run multiple copies of the SPF algorithm, one copy for each attached area. Area border routers condense the topological information of their attached areas for flooding to other areas.
- **Backbone routers**. A router that has an interface to the backbone. This includes all routers that interface to more than one area (i.e., area border routers). However, backbone routers do not have to be area border routers. Routers with all interfaces connected to the backbone are considered to be internal routers.
- **AS boundary routers**. A router that exchanges routing information with routers belonging to other Autonomous Systems. Such a router has AS external routes that are advertised throughout the Autonomous System. The path to each AS boundary router is known by every router in the AS. This classification is completely independent of the previous classifications (i.e., internal, area border, and backbone routers). AS boundary routers may be internal or area border routers, and may or may not participate in the backbone.

Virtual Links

It is possible to define areas in such a way that the backbone is no longer contiguous. (This is not an ideal OSPF configuration, and maximum effort should be made to avoid this situation.) In this case the system administrator must restore backbone connectivity by configuring *virtual links*.

Virtual links can be configured between any two backbone routers that have a connection to a common non-backbone area. The protocol treats two routers joined by a virtual link as if they were connected by an unnumbered point-to-point network. The routing protocol traffic that flows along the virtual link uses intra-area routing only, and the physical connection between the two routers is not managed by the network administrator (i.e., there is no dedicated connection between the routers as there is with the OSPF backbone).

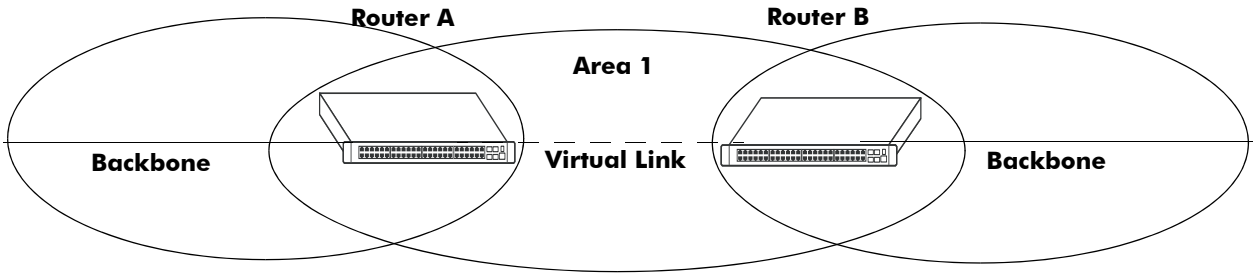


Figure 1-3 : OSPF Routers Connected with a Virtual Link

In the above diagram, Router A and Router B are connected via a virtual link in Area 1, which is known as a transit area. See “Creating Virtual Links” on page 1-21 for more information.

Stub Areas

OSPF allows certain areas to be configured as *stub areas*. A stub area is an area with routers that have no AS external Link State Advertisements (LSAs).

In order to take advantage of the OSPF stub area support, default routing must be used in the stub area. This is accomplished by configuring only one of the stub area's border routers to advertise a default route into the stub area. The default routes will match any destination that is not explicitly reachable by an intra-area or inter-area path (i.e., AS external destinations).

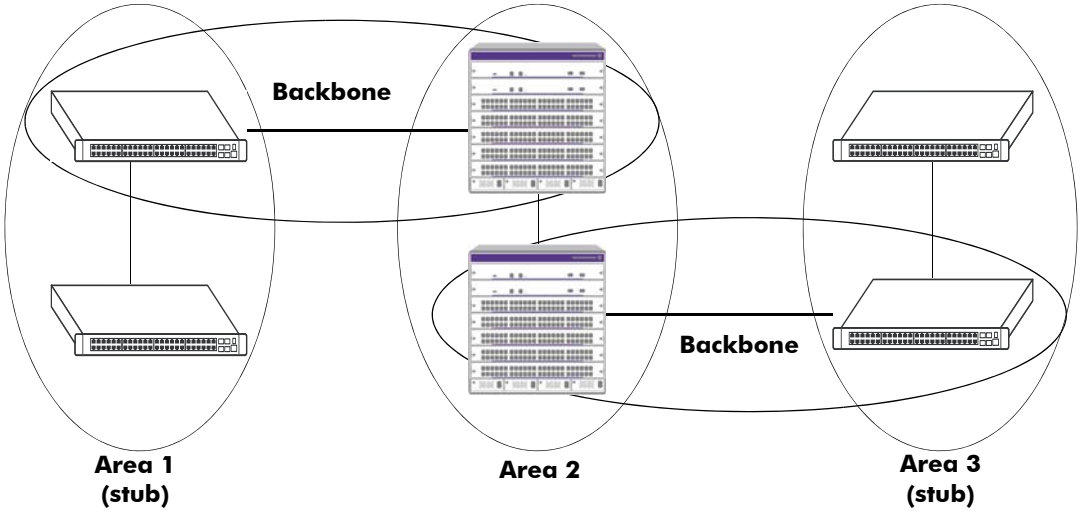


Figure 1-4 : OSPF Stub Area

Area 1 and Area 3 could be configured as stub areas. Stub areas are configured using the OSPF **ip ospf area** command, described in "Creating an Area" on page 1-15. For more overview information on areas, see "OSPF Areas" on page 1-7.

The OSPF protocol ensures that all routers belonging to an area agree on whether the area has been configured as a stub. This guarantees that no confusion will arise in the flooding of AS external advertisements.

Two restrictions on the use of stub areas are:

- Virtual links cannot be configured through stub areas.
- AS boundary routers cannot be placed internal to stub areas.

Not-So-Stubby-Areas

NSSA, or not-so-stubby area, is an extension to the base OSPF specification and is defined in RFC 1587. An NSSA is similar to a stub area in many ways: AS-external LSAs are not flooded into an NSSA and virtual links are not allowed in an NSSA. The primary difference is that selected external routing information can be imported into an NSSA and then redistributed into the rest of the OSPF routing domain. These routes are imported into the NSSA using a new LSA type: Type-7 LSA. Type-7 LSAs are flooded within the NSSA and are translated at the NSSA boundary into AS-external LSAs so as to convey the external routing information to other areas.

NSSAs enable routers with limited resources to participate in OSPF routing while also allowing the import of a selected number of external routes into the area. For example, an area which connects to a small external routing domain running RIP may be configured as an NSSA. This will allow the import of RIP routes into this area and the rest of the OSPF routing domain and at the same time, prevent the flooding of other external routing information (learned, for example, through RIP) into this area.

All routers in an NSSA must have their OSPF area defined as an NSSA. To configure otherwise will ensure that the router will be unsuccessful in establishing an adjacent in the OSPF domain.

Totally Stubby Areas

In Totally Stubby Areas the ABR advertises a default route to the routers in the totally stubby area but does not advertise any inter-area or external LSAs. As a result, routers in a totally stubby area know only the routes for destination networks in the stub area and have a default route for any other destination outside the stub.

Note. Virtual links cannot be configured through totally stubby areas.

The router memory is saved when using stub area networks by filtering Type 4 and 5 LSAs. This concept has been extended with Totally Stubby Areas by filtering Type 3 LSAs (Network Summary LSA) in addition to Type 4 and 5 with the exception of one single Type 3 LSA used to advertise a default route within the area.

The following is an example of a simple totally stubby configuration with Router B being an ABR between the backbone area 0 and the stub area 1. Router A is in area 1.1.1.1, totally stubby area:

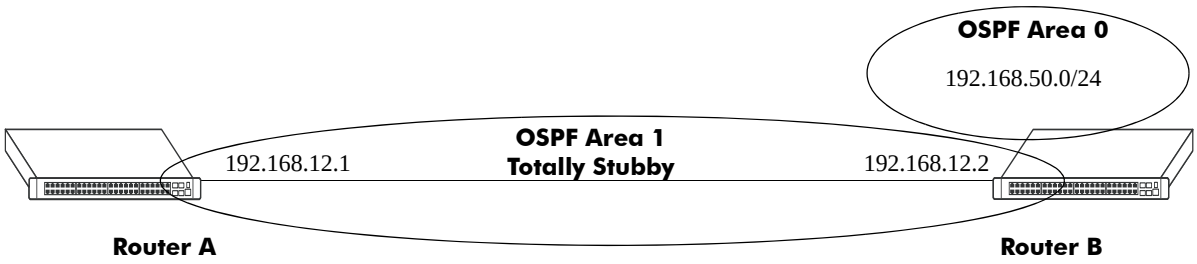


Figure 1-5 : Totally Stubby Area Example

Note. See “Configuring a Totally Stubby Area” on page 1-17 for information on configuring Totally Stubby Areas.

Equal Cost Multi-Path (ECMP) Routing

Using information from its continuously updated databases, OSPF calculates the shortest path to a given destination. Shortest path is determined from metric values at each hop along a path. At times, two or more paths to the same destination will have the same metric cost.

In the network illustration below, there are two paths from Source router A to Destination router B. One path traverses two hops at routers X and Y and the second path traverses two hops at M and N. If the total cost through X and Y to B is the same as the cost via M and N to B, then these two paths have equal cost. In this version of OSPF both paths will be stored and used to transmit data.

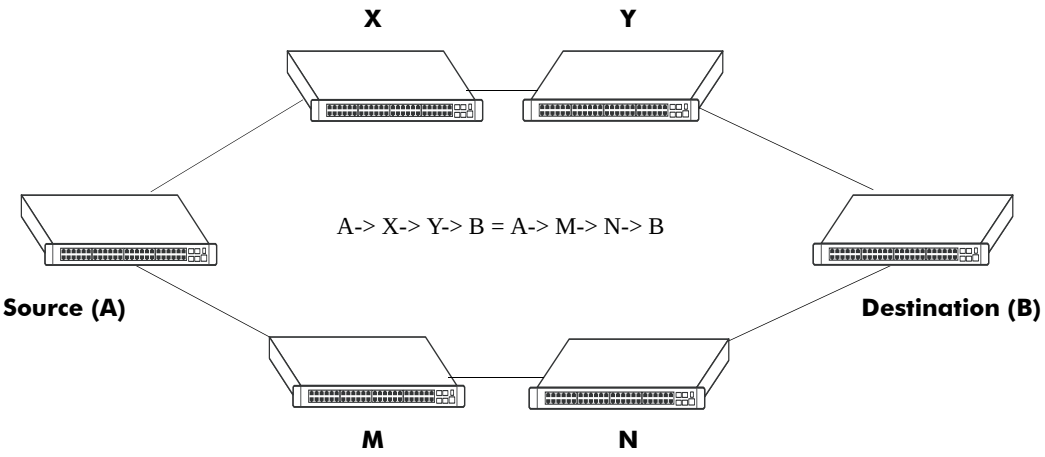


Figure 1-6 : Multiple Equal Cost Paths

Delivery of packets along equal paths is based on flows rather than a round-robin scheme. Equal cost is determined based on standard routing metrics. However, other variables, such as line speed, are not considered. So it is possible for OSPF to decide two paths have an equal cost even though one may contain faster links than another.

Non Broadcast OSPF Routing

OSPF can operate in two modes on non-broadcast networks: NBMA and point-to-multipoint. The interface type for the corresponding network segment should be set to non-broadcast or point-to-multipoint, respectively.

For non-broadcast networks neighbors should be statically configured. For NBMA neighbors the eligibility option must be enabled for the neighboring router to participate in Designated Router (DR) election.

For the correct working of an OSPF NBMA network, a fully meshed network is mandatory. Also, the neighbor eligibility configuration for a router on every other router should match the routers interface priority configuration.

See “Converting Local Interfaces into OSPF Passive Interface Using Route Map” on page 1-28 for more information and setting up static neighbors.

Graceful Restart on Switches with Redundant CMMs

A chassis-based switch with two Chassis management Modules (CMMs) can support redundancy where if the primary CMM fails or goes offline for any reason, the secondary CMM is instantly notified. The secondary CMM automatically assumes the primary role. This switch between the primary and secondary CMMs is known as *takeover*.

When a takeover occurs, which can be planned (e.g., the users performs the takeover) or unplanned (e.g., the primary CMM unexpectedly fails), an OSPF router must reestablish full adjacencies with all its previously fully adjacent neighbors. This time period between the restart and the reestablishment of adjacencies is termed *graceful restart*.

In the network illustration below, a helper router, Router Y, monitors the network for topology changes. As long as there are none, it continues to advertise its LSAs as if the restarting router, Router X, had remained in continuous OSPF operation (i.e., Router Y's LSAs continue to list an adjacency to Router X over network segment S, regardless of the adjacency's current synchronization state).

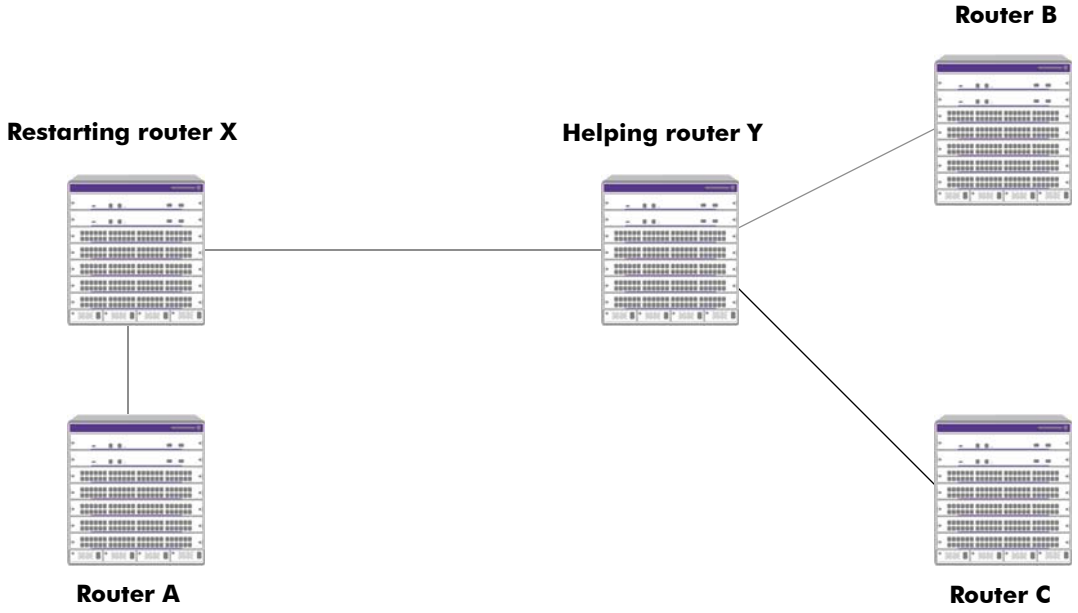


Figure 1-7 : OSPF Graceful Restart Helping and Restarting Router Example

If the restarting router, Router X, was the Designated Router (DR) on network segment S when the helping relationship began, the helper neighbor, Router Y, maintains Router X as the DR until the helping relationship is terminated. If there are multiple adjacencies with the restarting Router X, Router Y will act as a helper on all other adjacencies.

Note. See "Configuring Redundant CMMs for Graceful Restart" on page 1-30 for more information on configuring graceful restart.

Configuring OSPF

Configuring OSPF on a router requires several steps. Depending on your requirements, you may not need to perform all of the steps listed below.

By default, OSPF is disabled on the router. Configuring OSPF consists of these tasks:

- Set up the basics of the OSPF network by configuring the required VLANs, assigning ports to the VLANs, and assigning router identification numbers to the routers involved. This is described in "Preparing the Network for OSPF" on page 1-14.
- Enable OSPF. When the image file for advanced routing is installed, you must load the code and enable OSPF. The commands for enabling OSPF are described in "Activating OSPF" on page 1-14.
- Create an OSPF area and the backbone. The commands to create areas and backbones are described in "Creating an OSPF Area" on page 1-15.
- Set area parameters (optional). OSPF will run with the default area parameters, but different networks may benefit from modifying the parameters. Modifying area parameters is described in "Configuring Stub Area Default Metrics" on page 1-17.
- Create OSPF interfaces. OSPF interfaces are created and assigned to areas. Creating interfaces is described in "Creating an Interface" on page 1-18, and assigning interfaces is described in "Assigning an Interface to an Area" on page 1-18.
- Set interface parameters (optional). OSPF will run with the default interface parameters, but different networks may benefit from modifying the parameters. Also, it is possible to set authentication on an interface. Setting interface authentication is described in "Interface Authentication" on page 1-19, and modifying interface parameters is described in "Modifying Interface Parameters" on page 1-20.
- Configure virtual links (optional). A virtual link is used to establish backbone connectivity when two backbone routers are not physically contiguous. To create a virtual link, see "Creating Virtual Links" on page 1-21.
- Create a redistribution policy and enable the same using route maps (optional). To create route maps, see "Using Route Maps" on page 1-22.
- Configure router capabilities (optional). There are several commands that influence router operation. These are covered briefly in a table in "Configuring Router Capabilities" on page 1-27.
- Create static neighbors (optional). These commands allow you to statically configure neighbors. See "Converting Local Interfaces into OSPF Passive Interface Using Route Map" on page 1-28.
- Configure redundant switches for graceful OSPF restart (optional). Configuring switches with redundant switches for graceful restart is described in "Configuring Redundant CMMs for Graceful Restart" on page 1-30.
- Configure redundant CMMs for graceful OSPF restart (optional). Configuring switches with redundant switches for graceful restart is described in "Configuring Redundant CMMs for Graceful Restart" on page 1-30.

At the end of the chapter is a simple OSPF network diagram with instructions on how it was created on a router-by-router basis. See "OSPF Application Example" on page 1-31 for more information.

Preparing the Network for OSPF

OSPF operates on top of normal switch functions, using existing ports, virtual ports, VLANs, etc. The following network components should already be configured:

- **Configure VLANs that are to be used in the OSPF network**. VLANS should be created for both the backbone interfaces and all other connected devices that will participate in the OSPF network. A VLAN should exist for each instance in which the backbone connects two routers. VLAN configuration is described in "Configuring VLANs" in the *OmniSwitch AOS Release 8 Network Configuration Guide*.
- **Assign IP interfaces to the VLANs.** IP interfaces, or router ports, must be assigned to the VLAN. Assigning IP interfaces is described in "Configuring IP" in the *OmniSwitch AOS Release 8 Network Configuration Guide*.
- **Assign ports to the VLANs**. The physical ports participating in the OSPF network must be assigned to the created VLANs. Assigning ports to a VLAN is described in "Assigning Ports to VLANs" in the *OmniSwitch AOS Release 8 Network Configuration Guide*.
- **Set the router identification number**. (optional) The routers participating in the OSPF network must be assigned a router identification number. This number can be any number, as long as it is in standard dotted decimal format (e.g., 1.1.1.1). Router identification number assignment is discussed in "Configuring IP" in the *OmniSwitch AOS Release 8 Network Configuration Guide*. If this is not done, the router identification number is automatically the primary interface address.

Activating OSPF

To run OSPF on the router, the advanced routing image must be installed. For information on how to install image files, see the *OmniSwitch AOS Release 8 Switch Management Guide*.

After the image file has been installed onto the router, you will need to load the OSPF software into memory and enable it, as described below.

Loading the Software

To load the OSPF software into the router's running configuration, enter the **ip load ospf** command at the system prompt:

```
-> ip load ospf
```

The OPSF software is now loaded into memory, and can be enabled.

Enabling OSPF

Once the OSPF software has been loaded into the router's running configuration (either through the CLI or on startup), it must be enabled. To enable OSPF on a router, enter the **ip ospf admin-state** command at the CLI prompt, as shown:

```
-> ip ospf admin-state enable
```

Once OSPF is enabled, you can begin to set up OSPF parameters. To disable OSPF, enter the following:

```
-> ip ospf admin-state disable
```

Removing OSPF from Memory

To remove OSPF from the router memory, it is necessary to manually edit the **boot.cfg** file. The **boot.cfg** file is an ASCII text-based file that controls many of the switch parameters. Open the file and delete all references to OSPF.

For the operation to take effect the switch needs to be rebooted.

Creating an OSPF Area

OSPF allows a set of network devices in an AS system to be grouped together in *areas*.

There can be more than one router in an area. Likewise, there can be more than one area on a single router (in effect, making the router the Area Border Router (ABR) for the areas involved), but standard networking design does not recommended that more than three areas be handled on a single router.

Areas are named using 32-bit dotted decimal format (e.g., 1.1.1.1). Area 0.0.0.0 is reserved for the backbone.

Creating an Area

To create an area and associate it with a router, enter the **ip ospf area** command with the area identification number at the CLI prompt, as shown:

```
-> ip ospf area 1.1.1.1
```

Area 1.1.1.1 will now be created on the router with the default parameters.

The backbone is always area 0.0.0.0. To create this area on a router, you would use the above command, but specify the backbone, as shown:

```
-> ip ospf area 0.0.0.0
```

The backbone would now be attached to the router, making it an Area Border Router (ABR).

Specifying an Area Type

When creating areas, an area type can be specified (normal, stub, or NSSA). Area types are described above in “OSPF Areas” on page 1-7. To specify an area type, use the **ip ospf area** command as shown:

```
-> ip ospf area 1.1.1.1 type stub
```

Note. By default, an area is a **normal** area. The **type** keyword would be used to change a stub or NSSA area into a normal area.

Enabling and Disabling Summarization

Summarization can also be enabled or disabled when creating an area. Enabling summarization allows for ranges to be used by Area Border Routers (ABRs) for advertising routes as a single route rather than multiple routes, while disabling summarization prevents set ranges from functioning in stub and NSSA areas. (Configuring ranges is described in "Setting Area Ranges" on page 1-17.)

For example, to enable summarization for Area 1.1.1.1, enter the following:

```
-> ip ospf area 1.1.1.1 summary enable
```

To disable summarization for the same area, enter the following:

```
-> ip ospf area 1.1.1.1 summary disable
```

Note. By default, an area has summarization enabled. Disabling summarization for an area is useful when ranges need to be deactivated, but not deleted.

Displaying Area Status

You can check the status of the newly created area by using the **show** command, as demonstrated:

```
-> show ip ospf area 1.1.1.1
```

or

```
-> show ip ospf area
```

The first example gives specifics about area 1.1.1.1, and the second example shows all areas configured on the router.

To display a stub area's parameters, use the **show ip ospf area stub** command as follows:

```
-> show ip ospf area 1.1.1.1 stub
```

Deleting an Area

To delete an area, enter the **ip ospf area** command as shown:

```
-> no ip ospf area 1.1.1.1
```

Configuring Stub Area Default Metrics

The default metric configures the type of cost metric that a default area border router (ABR) will advertise in the default summary Link State Advertisement (LSA). Use the **ip ospf area default-metric** command to create or delete a default metric for stub or Not So Stubby Area (NSSA) area. Specify the stub area and select a cost value or a route type, as shown:

```
-> ip ospf area 1.1.1.1 default-metric 0 cost 50
```

or

```
-> ip ospf area 1.1.1.1 default-metric 0 type type1
```

A route has a preset metric associated to it depending on its type. The first example, the stub area is given a default metric of 0 (this is Type of Service 0) and a cost of 50 added to routes from the area. The second example specifies that the cost associated with Type 1 routes should be applied to routes from the area.

Note. At this time, only the default metric of ToS 0 is supported.

To remove the area default-metric setting, enter the **ip ospf area default-metric** command using the **no** command, as shown:

```
-> no ip ospf area 1.1.1.1 default-metric 0
```

Setting Area Ranges

Area ranges are used to summarize many area routes into a single advertisement at an area boundary. Ranges are advertised as summaries or NSSAs. Ranges also act as filters that either allow the summary to be advertised or not. Ranges are created using the **ip ospf area range** command. An area and the summary IP address and IP mask must be specified. For example, to create a summary range with IP address 192.5.40.1 and an IP mask of 255.255.255.0 for area 1.1.1.1, the following commands would be entered at the CLI prompt:

```
-> ip ospf area 1.1.1.1 range summary 192.5.40.1 255.255.255.0
-> ip ospf area 1.1.1.1 range summary 192.5.40.1 255.255.255.0 effect noMatching
```

To view the configured ranges for an area, use the **show ip ospf area range** command as demonstrated:

```
-> show ip ospf area 1.1.1.1 range
```

Configuring a Totally Stubby Area

In order to configure a totally stubby area you need to configure the area as stub on the ABR and disable summarization. By doing so the ABR will generate a default route in the totally stubby area. In addition, the other routers within the totally stubby area must only have their area configured as stub.

For example, to configure the simple totally stubby configuration shown in the figure in “Configuring a Totally Stubby Area” on page 1-17 where Router B is an ABR between the backbone area 0 and the stub area 1 and Router A is in Totally Stubby Area 1.1.1.1 follow the steps below:

1 Enter the following commands on Router B:

```
-> ip load ospf
-> ip ospf area 0.0.0.0
-> ip ospf area 1.1.1.1
-> ip ospf area 1.1.1.1 type stub
-> ip ospf area 1.1.1.1 summary disable
-> ip ospf area 1.1.1.1 default-metric 0
```

```
-> ip ospf interface vlan-5
-> ip ospf interface vlan-5 area 1.1.1.1
-> ip ospf interface vlan-5 admin-state enable
-> ip ospf interface vlan-6
-> ip ospf interface vlan-6 area 0.0.0.0
-> ip ospf interface vlan-6 admin-state enable
-> ip ospf admin-state enable
```

2 Enter the following on Router A:

```
-> ip load ospf
-> ip ospf area 1.1.1.1
-> ip ospf area 1.1.1.1 type stub
-> ip ospf interface vlan-3
-> ip ospf interface vlan-3 area 1.1.1.1
-> ip ospf interface vlan-3 admin-state enable
-> ip ospf admin-state enable
```

Creating OSPF Interfaces

Once areas have been established, interfaces need to be created and assigned to the areas.

Creating an Interface

To create an interface, enter the **ip ospf interface** command with an interface name, as shown:

```
-> ip ospf interface vlan-213
```

Note. The interface name *cannot* have spaces.

The interface can be deleted the by using the **no** keyword, as shown:

```
-> no ip ospf interface vlan-213
```

Assigning an Interface to an Area

Once an interface is created, it must be assigned to an area. (Creating areas is described in “Creating an Area” on page 1-15 above.)

To assign an interface to an area, enter the **ip ospf interface area** command with the interface name and area identification number at the CLI prompt. For example to add interface vlan-213 to area 1.1.1.1, enter the following:

```
-> ip ospf interface vlan-213 area 1.1.1.1
```

An interface can be removed from an area by reassigning it to a new area.

Once an interface has been created and enabled, you can check its status and configuration by using the **show ip ospf interface** command, as demonstrated:

```
-> show ip ospf interface vlan-213
```

Instructions for configuring authentication are given in “Interface Authentication” on page 1-19, and interface parameter options are described in “Modifying Interface Parameters” on page 1-20.

Activating an Interface

Once the interface is created and assigned to an area, it must be activated using the **ip ospf interface admin-state** command with the interface name, as shown:

```
-> ip ospf interface vlan-213 admin-state enable
```

The interface can be disabled using the **disable** keyword in place of the **enable** keyword.

Interface Authentication

OSPF allows authentication on the configured interfaces. When authentication is enabled, only neighbors using the same type of authentication and the matching passwords or keys can communicate.

There are three types of authentication: simple, MD5, and Keychain authentication. Simple authentication requires only a text string as a password, MD5 is a form of encrypted authentication that requires a key and a password, and a keychain is a form of authentication that allows a regular rotation of keys to be used for limited periods of time.

Simple Authentication

To enable simple authentication on an interface, enter the **ip ospf interface auth-type** command with the interface name, as shown:

```
-> ip ospf interface vlan-213 auth-type simple
```

Once simple authentication is enabled, the password must be set with the **ip ospf interface auth-key** command, as shown:

```
-> ip ospf interface vlan-213 auth-key test
```

In the above instance, only other interfaces with simple authentication and a password of “test” will be able to use the configured interface.

MD5 Encryption

To configure the same interface for MD5 encryption, enter the **ip ospf interface auth-type** as shown:

```
-> ip ospf interface vlan-213 auth-type md5
```

Once MD5 authentication is set, a key identification and key string must be set with the **ip ospf interface md5 key** command. For example to set interface 120.5.80.1 to use MD5 authentication with a key identification of 7 and key string of “test”, enter:

```
-> ip ospf interface vlan-213 md5 7
```

and

```
-> ip ospf interface vlan-213 md5 7 key "test"
```

Note that setting the key ID and key string must be done in two separate commands. Once the key ID and key string have been set, MD5 authentication is enabled. To disable it, use the **ip ospf interface md5** command, as shown:

```
-> ip ospf interface vlan-213 md5 7 disable
```

Keychain Authentication

To configure the same interface for keychain authentication, enter the **ip ospf interface auth-type** as shown:

```
-> ip ospf interface vlan-101 auth-type key-chain 1
```

When the OSPF interface receives a packet, the authentication information is carried in the hello packet. If the authentication succeeds, then adjacency is formed. The two remote machines must have the same active current key ID and same authentication type.

Use **show ip ospf interface** and **show ip ospf interface auth-info** commands to view the authentication information for the interface.

To remove all authentication, enter the **ip ospf interface auth-type** as follows:

```
-> ip ospf interface vlan-213 auth-type none
```

Modifying Interface Parameters

There are several interface parameters that can be modified on a specified interface. Most of these deal with timer settings.

The cost parameter and the priority parameter help to determine the cost of the route using this interface, and the chance that this interface's router will become the designated router, respectively.

The following table shows the various interface parameters that can be set:

ip ospf interface dead-interval	Configures OSPF interface dead interval. If no hello packets are received in this interval from a neighboring router the neighbor is considered dead.
ip ospf interface hello-interval	Configures the OSPF interface interval for NBMA segments.
ip ospf interface cost	Configures the OSPF interface cost. A cost metric refers to the network path preference assigned to certain types of traffic.
ip ospf interface poll-interval	Configures the OSPF poll interval.
ip ospf interface priority	Configures the OSPF interface priority. The priority number helps determine if this router will become the designated router.
ip ospf interface retrans-interval	Configures OSPF interface retransmit interval. The number of seconds between link state advertisement retransmissions for adjacencies belonging to this interface.
ip ospf interface transit-delay	Configures the OSPF interface transit delay. The estimated number of seconds required to transmit a link state update over this interface.

These parameters can be added any time. (See "Creating OSPF Interfaces" on page 1-18 for more information.) For example, to set an the dead interval to 50 and the cost to 100 on interface vlan-213, enter the following:

```
-> ip ospf interface vlan-213 dead-interval 50
-> ip ospf interface vlan-213 cost 100
```

To set an the poll interval to 25, the priority to 100, and the retransmit interval to 10 on interface vlan-213, enter the following:

```
-> ip ospf interface vlan-213 poll-interval 25 priority 100 retrans-interval 10
```

To set the hello interval to 5000 on interface vlan-213, enter the following:

```
-> ip ospf interface vlan-213 hello-interval 5000
```

To reset any parameter to its default value, enter the keyword with no parameter value, as shown:

```
-> ip ospf interface vlan-213 dead-interval
```

Note. Although you can configure several parameters at once, you can only reset them to the default one at a time.

Creating Virtual Links

A virtual link is a link between two backbones through a transit area. Use the **ip ospf virtual-link** command to create or delete a virtual link.

Accepted network design theory states that virtual links are the option of last resort. For more information on virtual links, see "Virtual Links" on page 1-8 and refer to the figure on page 1-8.

Creating a Virtual Link

To create a virtual link, commands must be submitted to the routers at both ends of the link. The router being configured should point to the other end of the link, and both routers must have a common area.

When entering the **ip ospf virtual-link** command, it is necessary to enter the Router ID of the far end of the link, and the area ID that both ends of the link share.

For example, a virtual link needs to be created between Router A (router ID 1.1.1.1) and Router B (router ID 2.2.2.2). We must:

1 Establish a transit area between the two routers using the commands discussed in "Creating an OSPF Area" on page 1-15 (in this example, we will use Area 0.0.0.1).

2 Then use the **ip ospf virtual-link** command on Router A as shown:

```
-> ip ospf virtual-link 0.0.0.1 2.2.2.2
```

3 Next, enter the following command on Router B:

```
-> ip ospf virtual-link 0.0.0.1 1.1.1.1
```

Now there is a virtual link across Area 0.0.0.1 linking Router A and Router B.

4 To display virtual links configured on a router, enter the following **show** command:

```
-> show ip ospf virtual-link
```

5 To delete a virtual link, enter the **ip ospf virtual-link** command with the area and far end router information, as shown:

```
-> no ip ospf virtual-link 0.0.0.1 2.2.2.2
```

Modifying Virtual Link Parameters

There are several parameters for a virtual link (such as authentication type and cost) that can be modified at the time of the link creation. They are described in the **ip ospf virtual-link** command description. These parameters are identical in function to their counterparts in the section "Modifying Interface Parameters" on page 1-20.

Using Route Maps

A route map specifies the criteria that are used to control redistribution of routes between protocols. Such criteria is defined by configuring route map statements. There are three different types of statements:

- **Action**. An action statement configures the route map name, sequence number, and whether or not redistribution is permitted or denied based on route map criteria.
- **Match.** A match statement specifies criteria that a route must match. When a match occurs, then the action statement is applied to the route.
- **Set.** A set statement is used to modify route information before the route is redistributed into the receiving protocol. This statement is only applied if all the criteria of the route map is met and the action permits redistribution.

The **ip route-map** command is used to configure route map statements and provides the following **action**, **match**, and **set** parameters:

ip route-map action ...	ip route-map match ...	ip route-map set ...
permit **deny**	**ip-address** **ip-nexthop** **ipv6-address** **ipv6-nexthop** **tag** **ipv4-interface** **ipv6-interface** **metric** **route-type**	**metric** **metric-type** **tag** **community** **local-preference** **level** **ip-nexthop** **ipv6-nexthop**

Refer to the “IP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the **ip route-map** command parameters and usage guidelines.

Once a route map is created, it is then applied using the **ip redist** command. See “Configuring Route Map Redistribution” on page 1-25 for more information.

Creating a Route Map

When a route map is created, it is given a name (up to 20 characters), a sequence number, and an action (permit or deny). Specifying a sequence number is optional.

To create a route map, use the **ip route-map** command with the **action** parameter. For example,

```
-> ip route-map ospf-to-bgp sequence-number 10 action permit
```

The above command creates the ospf-to-bgp route map, assigns a **sequence number** of 10 to the route map, and specifies a **permit** action.

To optionally filter routes before redistribution, use the **ip route-map** command with a **match** parameter to configure match criteria for incoming routes. For example,

```
-> ip route-map ospf-to-bgp sequence-number 10 match tag 8
```

The above command configures a match statement for the ospf-to-bgp route map to filter routes based on their tag value. When this route map is applied, only OSPF routes with a tag value of eight are redistributed into the BGP network. All other routes with a different tag value are dropped.

To modify route information before it is redistributed, use the **ip route-map** command with a **set** parameter. For example,

```
-> ip route-map ospf-to-bgp sequence-number 10 set tag 5
```

The above command configures a set statement for the ospf-to-bgp route map that changes the route tag value to five. Because this statement is part of the ospf-to-bgp route map, it is only applied to routes that have an existing tag value equal to eight.

The following is a summary of the commands used in the above examples:

```
-> ip route-map ospf-to-bgp sequence-number 10 action permit
-> ip route-map ospf-to-bgp sequence-number 10 match tag 8
-> ip route-map ospf-to-bgp sequence-number 10 set tag 5
```

To verify a route map configuration, use the **show ip route-map** command:

```
-> show ip route-map
Route Maps: configured: 1 max: 200
Route Map: ospf-to-bgp Sequence Number: 10 Action permit
  match tag 8
  set tag 5
```

Deleting a Route Map

Use the **no** form of the **ip route-map** command to delete an entire route map, a route map sequence, or a specific statement within a sequence.

To delete an entire route map, enter **no ip route-map** followed by the route map name. For example, the following command deletes the entire route map named redistipv4:

```
-> no ip route-map redistipv4
```

To delete a specific sequence number within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the actual number. For example, the following command deletes sequence 10 from the redistipv4 route map:

```
-> no ip route-map redistipv4 sequence-number 10
```

Note that in the above example, the redistripv4 route map is not deleted. Only those statements associated with sequence 10 are removed from the route map.

To delete a specific statement within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the sequence number for the statement, then either **match** or **set** and the match or set parameter and value. For example, the following command deletes only the match tag 8 statement from route map redistipv4 sequence 10:

```
-> no ip route-map redistipv4 sequence-number 10 match tag 8
```

Setting the Metric

A route map can be used to set the metric by adding, subtracting, or replacing the metric of a route as in the example below:

```
-> ip route-map set-metric set metric 1 effect add
```

- Add - Adds the given value to the routes metric
- Subtract - Subtracts the given value from the metric (can't be less than 1)
- Replace - Uses the given value for the routes metric
- None - Ignores the given value and passes the routes metric through

Denying A Route

With route maps denying a route does not mean that all the other routes are automatically permitted. It is necessary to configure proper permit/deny rule for each route. However, a permit rule can be be created to allow all routes and then specific rules for denying certain routes can be created as in the example below:

```
-> ip route-map leakin match ip-address 0.0.0.0/0 permit (permits all routes)
-> ip route-map leakin-example match ip-address 14.14.0.0/16 (deny route)
```

Configuring Route Map Sequences

A route map may consist of one or more sequences of statements. The sequence number determines which statements belong to which sequence and the order in which sequences for the same route map are processed.

To add match and set statements to an existing route map sequence, specify the same route map name and sequence number for each statement. For example, the following series of commands creates route map rm_1 and configures match and set statements for the rm_1 sequence 10:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 set metric 1
```

To configure a new sequence of statements for an existing route map, specify the same route map name but use a different sequence number. For example, the following command creates a new sequence 20 for the rm_1 route map:

```
-> ip route-map rm_1 sequence-number 20 action permit
-> ip route-map rm_1 sequence-number 20 match ipv4-interface to-finance
-> ip route-map rm_1 sequence-number 20 set metric 5
```

The resulting route map appears as follows:

```
-> show ip route-map rm_1
Route Map: rm_1 Sequence Number: 10 Action permit
  match tag 8
  set metric 1
Route Map: rm_1 Sequence Number: 20 Action permit
  match ip6 interface to-finance
  set metric 5
```

Sequence 10 and sequence 20 are both linked to route map rm_1 and are processed in ascending order according to their sequence number value. Note that there is an implied logical OR between sequences. As a result, if there is no match for the tag value in sequence 10, then the match interface statement in sequence 20 is processed. However, if a route matches the tag 8 value, then sequence 20 is not used. The set statement for whichever sequence was matched is applied.

A route map sequence may contain multiple match statements. If these statements are of the same kind (e.g., match tag 5, match tag 8, etc.) then a logical OR is implied between each like statement. If the match statements specify different types of matches (e.g. match tag 5, match ip4 interface to-finance, etc.), then a logical AND is implied between each statement. For example, the following route map sequence will redistribute a route if its tag is either 8 or 5:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
```

The following route map sequence will redistribute a route if the route has a tag of 8 or 5 *and* the route was learned on the IPv4 interface to-finance:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 match ipv4-interface to-finance
```

Configuring Access Lists

An IP access list provides a convenient way to add multiple IPv4 or IPv6 addresses to a route map. Using an access list avoids having to enter a separate route map statement for each individual IP address. Instead, a single statement is used that specifies the access list name. The route map is then applied to all the addresses contained within the access list.

Configuring an IP access list involves two steps: creating the access list and adding IP addresses to the list. To create an IP access list, use the **ip access-list** command (IPv4) or the **ipv6 access-list** command (IPv6) and specify a name to associate with the list. For example,

```
-> ip access-list ipaddr
-> ipv6 access-list ip6addr
```

To add addresses to an access list, use the **ip access-list address** (IPv4) or the **ipv6 access-list address** (IPv6) command. For example, the following commands add addresses to an existing access list:

```
-> ip access-list ipaddr address 16.24.2.1/16
-> ipv6 access-list ip6addr address 2001::1/64
```

Use the same access list name each time the above commands are used to add additional addresses to the same access list. In addition, both commands provide the ability to configure if an address and/or its matching subnet routes are permitted (the default) or denied redistribution. For example:

```
-> ip access-list ipaddr address 16.24.2.1/16 action deny redist-control all-
subnets
-> ipv6 access-list ip6addr address 2001::1/64 action permit redist-control no-
subnets
```

For more information about configuring access list commands, see the “IP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Route Map Redistribution

The **ip redist** command is used to configure the redistribution of routes from a source protocol into the destination protocol. This command is used on the router that will perform the redistribution.

Note. An OSPF router automatically becomes an Autonomous System Border Router (ASBR) when redistribution is configured on the router.

A source protocol is a protocol from which the routes are learned. A destination protocol is the one into which the routes are redistributed. Make sure that both protocols are loaded and enabled before configuring redistribution.

Redistribution applies criteria specified in a route map to routes received from the source protocol. Therefore, configuring redistribution requires an existing route map. For example, the following command configures the redistribution of OSPF routes into the BGP network using the ospf-to-bgp route map:

```
-> ip redist ospf into bgp route-map ospf-to-bgp
```

OSPF routes received by the router interface are processed based on the contents of the ospf-to-bgp route map. Routes that match criteria specified in this route map are either allowed or denied redistribution into the BGP network. The route map may also specify the modification of route information before the route

is redistributed. See "Using Route Maps" on page 1-22 for more information.

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

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

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

```
-> show ip redist

Source       Destination
Protocol     Protocol     Status    Route Map
-------------+------------+---------+--------------------
LOCAL4       RIP          Enabled   rip_1
LOCAL4       OSPF         Enabled   ospf_2
LOCAL4       BGP          Enabled   bgp_3
BGP          OSPF         Enabled   ospf-to-bgp
```

Configuring the Administrative Status of the Route Map Redistribution

The administrative status of a route map redistribution configuration is enabled by default. To change the administrative status, use the **status** parameter with the **ip redist** command. For example, the following command disables the redistribution administrative status for the specified route map:

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

The following command example enables the administrative status:

```
-> ip redist ospf into bgp route-map ospf-to-bgp admin-state enable
```

Route Map Redistribution Example

The following example configures the redistribution of OSPF routes into a BGP network using a route map (ospf-to-bgp) to filter specific routes:

```
-> ip route-map ospf-to-bgp sequence-number 10 action deny
-> ip route-map ospf-to-bgp sequence-number 10 match tag 5
-> ip route-map ospf-to-bgp sequence-number 10 match route-type external type2
-> ip route-map ospf-to-bgp sequence-number 20 action permit
-> ip route-map ospf-to-bgp sequence-number 20 match ipv4-interface intf_ospf
-> ip route-map ospf-to-bgp sequence-number 20 set metric 255

-> ip route-map ospf-to-bgp sequence-number 30 action permit
-> ip route-map ospf-to-bgp sequence-number 30 set tag 8

-> ip redist ospf into bgp route-map ospf-to-bgp
```

The resulting ospf-to-bgp route map redistribution configuration does the following:

- Denies the redistribution of Type 2 external OSPF routes with a tag set to five.
- Redistributes into BGP all routes learned on the intf_ospf interface and sets the metric for such routes to 255.
- Redistributes into BGP all other routes (those not processed by sequence 10 or 20) and sets the tag for such routes to eight.

Configuring Router Capabilities

The following list shows various commands that can be useful in tailoring a router's performance capabilities. All of the listed parameters have defaults that are acceptable for running an OSPF network.

ip ospf exit-overflow-interval	Sets the overflow interval value. The overflow interval is the time whereby the router will wait before attempting to leave the database overflow state.
ip ospf extlsdb-limit	Sets a limit to the number of external Link State Databases entries learned by the router. An external LSDB entry is created when the router learns a link address that exists outside of its Autonomous System (AS).
ip ospf host	Creates and deletes an OSPF entry for directly attached hosts.
ip ospf mtu-checking	Enables or disables the use of Maximum Transfer Unit (MTU) checking on received OSPF database description packets.
ip ospf default-originate	Configures a default external route into the OSPF routing domain.
ip ospf route-tag	Configures a tag value for OSPF routes injected into the IP routing table that can be used for redistribution.
ip ospf spf-timer	Configures timers for Shortest Path First (SPF) calculation.

To configure a router parameter, enter the parameter at the CLI prompt with the new value or required variables. For example to set the exit overflow interval to 40, enter:

```
-> ip ospf exit-overflow-interval 40
```

To enable MTU checking, enter:

```
-> ip ospf mtu-checking
```

To advertise a default external route into OSPF regardless of whether the routing table has a default route, enter:

```
-> ip ospf default-originate always
```

To set the route tag to 5, enter:

```
-> ip ospf route-tag 5
```

To set the SPF timer delay to 3 and the hold time to 6, enter:

```
-> ip ospf spf-timer delay 3 hold 6
```

To return a parameter to its default setting, enter the command with no parameter value, as shown:

```
-> ip ospf spf-timer
```

Converting Local Interfaces into OSPF Passive Interface Using Route Map

Passive interfaces do not accept or send routing updates. In an OSPF network, an interface can be configured as passive (by setting the hello interval and dead interval to “0”) mainly to add this interface in the updates to the OSPF neighbor. No OSPF adjacency is formed on a passive interface, and if a OSPF-enabled interface is configured as passive where an adjacency already exists, the adjacency drops almost immediately.

In a scenario where there is a requirement to configure more number of passive OSPF interfaces in an Area, route map can be used. A route map with set action of route-type ‘internal’ needs to be created for the local interface (routes) on which passive OSPF interface needs to be created. Using this route map in redistribution, any or all local interfaces can be converted into passive OSPF interfaces.

Example:

Include the IP interfaces which need to be configured as passive OSPF interface in a route map, and set metric type as ‘internal’ and then redistribute ‘local into ospf’.

```
-> ip route-map "R1" action permit
-> ip route-map "R1" match ip-address 10.10.0.0/16
-> ip route-map "R1" set metric-type internal
-> ip redist local into ospf route-map R1 admin-state enable
```

For more information about configuring route map and the other related commands, see the “IP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* and “Configuring IP” chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

Note. If there are multiple areas configured in an OSPF domain, the passive OSPF interfaces will be created in the area with the lowest-numbered Area ID, which is usually the Backbone Area.

Use **show ip ospf interface** command to view the passive OSPF interfaces.

Configuring Static Neighbors

It is possible to configure neighbors statically on Non Broadcast Multi Access (NBMA), point-to-point, and point-to-multipoint networks.

NBMA requires all routers attached to the network to communicate directly (unicast), and every attached router in this network becomes aware of all of its neighbors through configuration. It also requires a Designated Router (DR) “eligibility” flag to be set for every neighbor.

To set up a router to use NBMA routing, follow the following steps:

1 Create an OSPF interface using the CLI command **ip ospf interface** and perform all the normal configuration for the interface as with broadcast networks (attaching it to an area, enabling the status, etc.).

2 The OSPF interface type for this interface should be set to non-broadcast using the CLI **ip ospf interface type** command. For example, to set interface vlan-213 to be an NBMA interface, enter the following:

```
-> ip ospf interface vlan-213 type non-broadcast
```

3 Configure static neighbors for every OSPF router in the network using the **ip ospf neighbor** command. For example, to create an OSPF neighbor with an IP address of 1.1.1.8 to be a static neighbor, enter the following:

```
-> ip ospf neighbor 1.1.1.8 eligible
```

The neighbor attaches itself to the right interface by matching the network address of the neighbor and the interface. If the interface has not yet been created, the neighbor gets attached to the interface as and when the interface comes up.

If this neighbor is not required to participate in DR election, configure it as ineligible. The eligibility can be changed at any time as long as the interface it is attached to is in the disabled state.

Configuring Redundant CMMs for Graceful Restart

By default, OSPF graceful restart is disabled. To enable OSPF graceful restart on OmniSwitch chassis-based switches, use the **ip ospf restart-support** command by entering ip **ospf restart-support** followed by **planned-unplanned**.

For example, to enable OSPF graceful restart to support planned and unplanned restarts enter:

```
-> ip ospf restart-support planned-unplanned
```

To disable OSPF graceful restart use the **no** form of the **ip ospf restart-support** command by entering:

```
-> no ip ospf restart-support
```

Optionally, you can configure graceful restart parameters with the following CLI commands:

ip ospf restart-interval	Configures the grace period for achieving a graceful OSPF restart.
ip ospf restart-helper admin-state	Administratively enables and disables the capability of an OSPF router to operate in helper mode in response to a router performing a graceful restart.
ip ospf restart-helper strict-lsa-checking admin-state	Administratively enables and disables whether or not a changed Link State Advertisement (LSA) will result in termination of graceful restart by a helping router.
ip ospf restart initiate	Initiates a planned graceful restart.

For more information about graceful restart commands, see the “OSPF Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Redistribution of Internal BGP Routes to OSPF

By default, redistribution of iBGP routes is not allowed into OSPF protocol. To allow redistribution of iBGP routes (from the same AS) into OSPF protocol, use the **ip ospf redist-bgp-internal** command.

```
-> ip ospf redist-bgp-internal
```

To disable redistribution of iBGP routes into OSPF protocol use the **no** form of the **ip ospf redist-bgp-internal** command by entering:

```
-> no ip ospf redist-bgp-internal
```

OSPF Application Example

This section will demonstrate how to set up a simple OSPF network. It uses three routers, each with an area. Each router uses three VLANs. A backbone connects all the routers. This section will demonstrate how to set it up by explaining the necessary commands for each router.

The following diagram is a simple OSPF network. It will be created by the steps listed on the following pages:

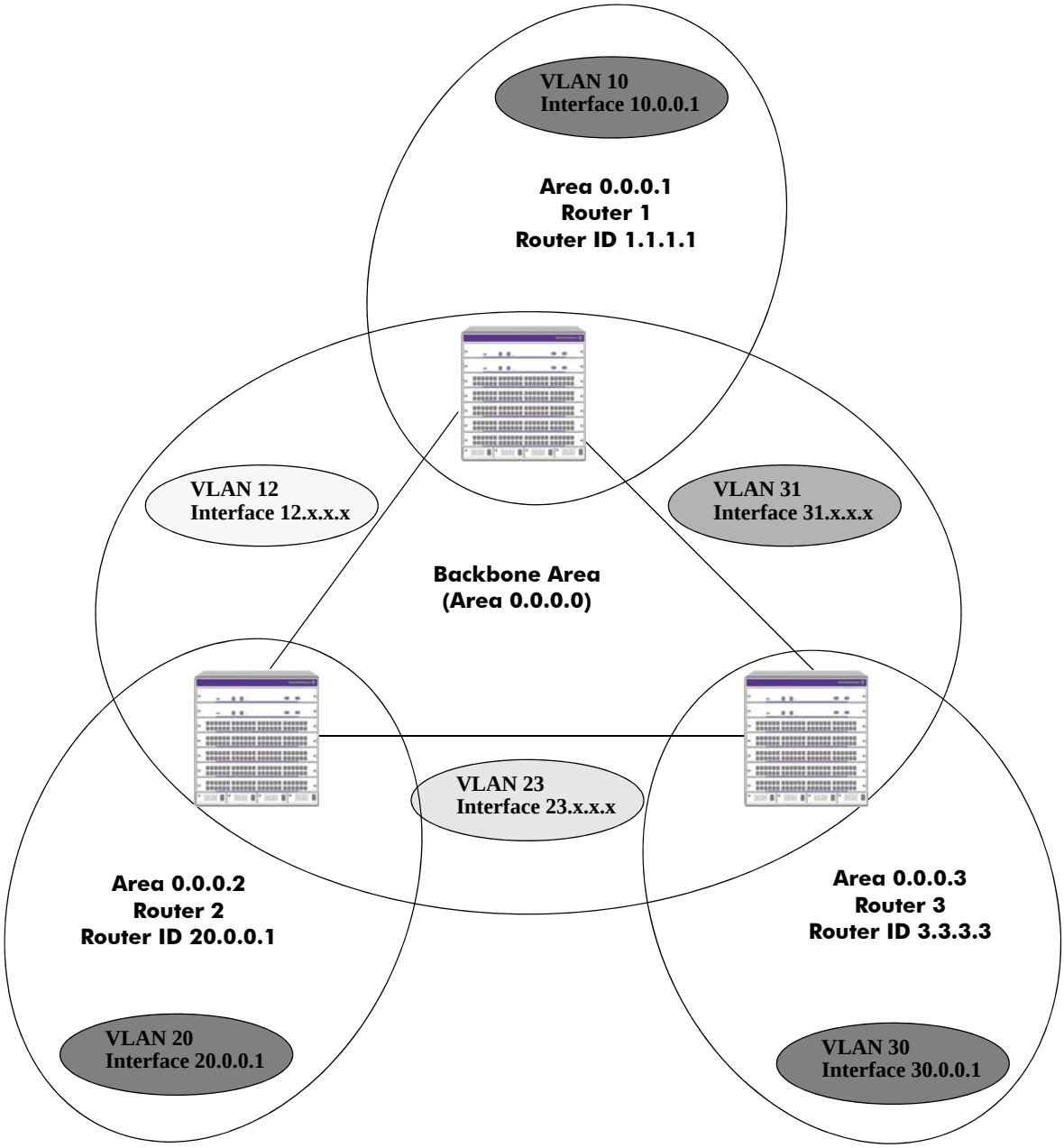


Figure 1-8 : Three Area OSPF Network

Step 1: Prepare the Routers

The first step is to create the VLANs on each router, add an IP interface to the VLAN, assign a port to the VLAN, and assign a router identification number to the routers. For the backbone, the network design in this case uses slot 2, port 1 as the egress port and slot 2, port 2 as ingress port on each router. Router 1 connects to Router 2, Router 2 connects to Router 3, and Router 3 connects to Router 1 using 10/100 Ethernet cables.

Note. The ports will be statically assigned to the router, as a VLAN must have a physical port assigned to it in order for the router port to function. However, the router could be set up in such a way that mobile ports are dynamically assigned to VLANs using VLAN rules. See the chapter titled "Defining VLAN Rules" in the *OmniSwitch AOS Release 8 Network Configuration Guide.*

The commands setting up VLANs are shown below:

Router 1 (using ports 2/1 and 2/2 for the backbone, and ports 2/3-5 for end devices):

```
-> vlan 31
-> ip interface vlan-31 vlan 31 address 31.0.0.1 mask 255.0.0.0
-> vlan 31 members port 2/1 untagged

-> vlan 12
-> ip interface vlan-12 vlan 12 address 12.0.0.1 mask 255.0.0.0
-> vlan 12 members port 2/2 untagged

-> vlan 10
-> ip interface vlan-10 vlan 10 address 10.0.0.1 mask 255.0.0.0
-> vlan 10 members port 2/3-5 untagged

-> ip router router-id 1.1.1.1
```

These commands created VLANs 31, 12, and 10.

- VLAN 31 handles the backbone connection from Router 1 to Router 3, using the IP router port 31.0.0.1 and physical port 2/1.
- VLAN 12 handles the backbone connection from Router 1 to Router 2, using the IP router port 12.0.0.1 and physical port 2/2.
- VLAN 10 handles the device connections to Router 1, using the IP router port 10.0.0.1 and physical ports 2/3-5. More ports could be added at a later time if necessary.

The router was assigned the Router ID of 1.1.1.1.

Router 2 (using ports 2/1 and 2/2 for the backbone, and ports 2/3-5 for end devices):

```
-> vlan 12
-> ip interface vlan-12 vlan 12 address 12.0.0.2 mask 255.0.0.0
-> vlan 12 members port 2/1 untagged

-> vlan 23
-> ip interface vlan-23 vlan 23 address 23.0.0.2 mask 255.0.0.0
-> vlan 23 members port 2/2 untagged

-> vlan 20
-> ip interface vlan-20 vlan 20 address 20.0.0.2 mask 255.0.0.0
-> vlan 20 members port 2/3-5 untagged

-> ip router router-id 2.2.2.2
```

These commands created VLANs 12, 23, and 20.

- VLAN 12 handles the backbone connection from Router 1 to Router 2, using the IP router port 12.0.0.2 and physical port 2/1.
- VLAN 23 handles the backbone connection from Router 2 to Router 3, using the IP router port 23.0.0.2 and physical port 2/2.
- VLAN 20 handles the device connections to Router 2, using the IP router port 20.0.0.2 and physical ports 2/3-5. More ports could be added at a later time if necessary.

The router was assigned the Router ID of 2.2.2.2.

Router 3 (using ports 2/1 and 2/2 for the backbone, and ports 2/3-5 for end devices):

```
-> vlan 23
-> ip interface vlan-23 vlan 23 address 23.0.0.3 mask 255.0.0.0
-> vlan 23 members port 2/1 untagged

-> vlan 31
-> ip interface vlan-31 vlan 31 address 31.0.0.3 mask 255.0.0.0
-> vlan 31 members port 2/2 untagged

-> vlan 30
-> ip interface vlan-30 vlan 30 address 30.0.0.3 mask 255.0.0.0
-> vlan 30 members port 2/3-5 untagged

-> ip router router-id 3.3.3.3
```

These commands created VLANs 23, 31, and 30.

- VLAN 23 handles the backbone connection from Router 2 to Router 3, using the IP router port 23.0.0.3 and physical port 2/1.
- VLAN 31 handles the backbone connection from Router 3 to Router 1, using the IP router port 31.0.0.3 and physical port 2/2.
- VLAN 30 handles the device connections to Router 3, using the IP router port 30.0.0.3 and physical ports 2/3-5. More ports could be added at a later time if necessary.

The router was assigned the Router ID of 3.3.3.3.

Step 2: Enable OSPF

The next step is to load and enable OSPF on each router. The commands for this step are below (the commands are the same on each router):

```
-> ip load ospf
-> ip ospf admin-state enable
```

Step 3: Create the Areas and Backbone

Now the areas should be created. In this case, we will create an area for each router, and a backbone (area 0.0.0.0) that connects the areas.

The commands for this step are below:

Router 1

```
-> ip ospf area 0.0.0.0
-> ip ospf area 0.0.0.1
```

These commands created area 0.0.0.0 (the backbone) and area 0.0.0.1 (the area for Router 1). Both of these areas are also enabled.

Router 2

```
-> ip ospf area 0.0.0.0
-> ip ospf area 0.0.0.2
```

These commands created Area 0.0.0.0 (the backbone) and Area 0.0.0.2 (the area for Router 2). Both of these areas are also enabled.

Router 3

```
-> ip ospf area 0.0.0.0
-> ip ospf area 0.0.0.3
```

These commands created Area 0.0.0.0 (the backbone) and Area 0.0.0.3 (the area for Router 3). Both of these areas are also enabled.

Step 4: Create, Enable, and Assign Interfaces

Next, OSPF interfaces must be created, enabled, and assigned to the areas. The OSPF interfaces should have the same interface name as the IP router ports created above in "Step 1: Prepare the Routers" on page 1-32.

Router 1

```
-> ip ospf interface vlan-31
-> ip ospf interface vlan-31 area 0.0.0.0
-> ip ospf interface vlan-31 admin-state enable

-> ip ospf interface vlan-12
-> ip ospf interface vlan-12 area 0.0.0.0
-> ip ospf interface vlan-12 admin-state enable

-> ip ospf interface  vlan-10
-> ip ospf interface  vlan-10 area 0.0.0.1
-> ip ospf interface  vlan-10 admin-state enable
```

IP router port 31.0.0.1 was associated to OSPF interface vlan-31, enabled, and assigned to the backbone. IP router port 12.0.0.1 was associated to OSPF interface vlan-12, enabled, and assigned to the backbone. IP router port 10.0.0.1 which connects to end stations and attached network devices, was associated to OSPF interface vlan-10, enabled, and assigned to Area 0.0.0.1.

Router 2

```
-> ip ospf interface vlan-12
-> ip ospf interface vlan-12 area 0.0.0.0
-> ip ospf interface vlan-12 admin-state enable

-> ip ospf interface vlan-23
-> ip ospf interface vlan-23 area 0.0.0.0
-> ip ospf interface vlan-23 admin-state enable
```

```
-> ip ospf interface vlan-20
-> ip ospf interface vlan-20 area 0.0.0.2
-> ip ospf interface vlan-20 admin-state enable
```

IP router port 12.0.0.2 was associated to OSPF interface vlan-12, enabled, and assigned to the backbone.
IP router port 23.0.0.2 was associated to OSPF interface vlan-23, enabled, and assigned to the backbone.
IP router port 20.0.0.2, which connects to end stations and attached network devices, was associated to OSPF interface vlan-20, enabled, and assigned to Area 0.0.0.2.

Router 3

```
-> ip ospf interface vlan-23
-> ip ospf interface vlan-23 area 0.0.0.0
-> ip ospf interface vlan-23 admin-state enable

-> ip ospf interface vlan-31
-> ip ospf interface vlan-31 area 0.0.0.0
-> ip ospf interface vlan-31 admin-state enable

-> ip ospf interface vlan-30
-> ip ospf interface vlan-30 area 0.0.0.3
-> ip ospf interface vlan-30 admin-state enable
```

IP router port 23.0.0.3 was associated to OSPF interface vlan-23, enabled, and assigned to the backbone.
IP router port 31.0.0.3 was associated to OSPF interface vlan-31, enabled, and assigned to the backbone.
IP router port 30.0.0.3, which connects to end stations and attached network devices, was associated to OSPF interface vlan-30, enabled, and assigned to Area 0.0.0.3.

Step 5: Examine the Network

After the network has been created, you can check various aspects of it using show commands:

- For OSPF in general, use the **show ip ospf** command.
- For areas, use the **show ip ospf area** command.
- For interfaces, use the **show ip ospf interface** command.
- To check for adjacencies formed with neighbors, use the **show ip ospf neighbor** command.
- For routes, use the **show ip ospf routes** command.

Verifying OSPF Configuration

To display information about areas, interfaces, virtual links, redistribution, or OPSF in general, use the **show** commands listed in the following table:

show ip ospf	Displays OSPF status and general configuration parameters.
show ip ospf border-routers	Displays information regarding all or specified border routers.
show ip ospf ext-lsdb	Displays external Link State Advertisements from the areas to which the router is attached.
show ip ospf host	Displays information on directly attached hosts.
show ip ospf lsdb	Displays LSAs in the Link State Database associated with each area.
show ip ospf neighbor	Displays information on OSPF non-virtual neighbor routers.
show ip redist	Displays the route map redistribution configuration.
show ip ospf routes	Displays OSPF routes known to the router.
show ip ospf virtual-link	Displays virtual link information.
show ip ospf virtual-neighbor	Displays OSPF virtual neighbors.
show ip ospf area	Displays either all OSPF areas, or a specified OSPF area.
show ip ospf area range	Displays all or specified configured area address range summaries for the given area.
show ip ospf area stub	Displays stub area status.
show ip ospf interface	Displays OSPF interface information.
show ip ospf restart	Displays the OSPF graceful restart related configuration and status.

For more information about the resulting displays form these commands, see the “OSPF Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Examples of the **show ip ospf**, **show ip ospf area**, and **show ip ospf interface** command outputs are given in the section “OSPF Quick Steps” on page 1-3.

2 Configuring OSPFv3

Open Shortest Path First version 3 (OSPFv3) is an extension of OSPF version 2 that provides support for networks using the IPv6 protocol. OSPFv2 is for IPv4 networks (see Chapter 1, “Configuring OSPF,” for more information about OSPFv2).

In This Chapter

This chapter describes the basic components of OSPFv3 and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Loading and enabling OSPFv3. See “Activating OSPFv3” on page 2-15.
- Creating OSPFv3 areas. See “Creating an OSPFv3 Area” on page 2-16.
- Creating OSPFv3 interfaces. See “Creating OSPFv3 Interfaces” on page 2-19.
- Creating virtual links. See “Creating Virtual Links” on page 2-21.
- Configuring redistribution using route map. See “Configuring Redistribution” on page 2-21.
- Configuring router capabilities. See “Configuring Router Capabilities” on page 2-27.
- Configuring static neighbors. See “Configuring Static Neighbors” on page 2-28.
- Configuring redundant CMMs for graceful restart. See “Configuring Redundant CMMs for Graceful Restart” on page 2-29.

For information on creating and managing VLANs, see “Configuring VLANs” in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

OSPFv3 Defaults Table

The following table shows the default settings of the configurable OSPFv3 parameters.

Parameter Description	Command	Default Value/Comments
Configures the OSPFv3 administrative status.	**ipv6 ospf admin-state**	enabled
Configures the administrative status for an OSPF interface.	**ipv6 ospf interface admin-state**	enabled
Configures OSPFv3 redistribution.	**ipv6 redist**	disabled
Configures timers for Shortest Path First (SPF) calculation.	**ipv6 ospf spf-timer**	delay: 5 hold: 10
Creates or deletes an area default metric.	**ipv6 ospf area**	0
Configures OSPFv3 interface dead interval.	**ipv6 ospf interface dead-interval**	40 seconds
Configures OSPFv3 interface hello interval.	**ipv6 ospf interface hello-interval**	10 seconds
Configures the OSPFv3 interface cost.	**ipv6 ospf interface cost**	1
Configures the OSPFv3 interface priority.	**ipv6 ospf interface priority**	1
Configures OSPFv3 interface retransmit interval.	**ipv6 ospf interface retrans-interval**	5 seconds
Configures the OSPFv3 interface transit delay.	**ipv6 ospf interface transit-delay**	1 second
Configures support for the graceful restart feature on an OSPFv3 router.	**ipv6 ospf restart**	enabled

OSPFv3 Quick Steps

The followings steps are designed to show the user the necessary set of commands for setting up a router to use OSPFv3:

1 Create a VLAN using the **vlan** command. For example:

```
-> vlan 5
-> vlan 5 admin-state enable
```

2 Create an IPv6 interface on the vlan using the **ipv6 interface** command. For example:

```
-> ipv6 interface test vlan 1
```

3 Configure an IPv6 address on the vlan using the **ipv6 address** command. For example:

```
-> ipv6 address 2001::/64 eui-64 test
```

4 Assign a port to the VLAN created in Step 1 using the **vlan members** command. For example:

```
-> vlan 1 members port 2/1 untagged
```

Note. The port will be statically assigned to the VLAN, as a VLAN must have a physical port assigned to it in order for the router port to function. However, the router could be set up in such a way that mobile ports are dynamically assigned to VLANs using VLAN rules. See the chapter titled “Defining VLAN Rules” in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

---

5 Assign a router ID to the router using the **ip router router-id** command. For example:

```
-> ip router router-id 5.5.5.5
```

6 Load and enable OSPFv3 using **the ipv6 load ospf** and the **ipv6 ospf admin-state** commands.

```
-> ipv6 load ospf
-> ipv6 ospf admin-state enable
```

7 Create a backbone to connect this router to others, and an area for the router’s traffic using the **ipv6 ospf area** command. (Backbones are always labeled area 0.0.0.0.) For example:

```
-> ipv6 ospf area 0.0.0.0
-> ipv6 ospf area 0.0.0.1
```

8 Create an OSPFv3 interface for the VLAN created in Step 1 and assign the interface to an area identifier using the **ipv6 ospf interface area** command. The OSPFv3 interface should use the same interface name used for the VLAN router IP created in Step 2. For example:

```
-> ipv6 ospf interface test area 0.0.0.0
```

Note. The interface name *cannot* have spaces.

---

9 Enable the OSPFv3 interface using the **ipv6 ospf interface admin-state** command. For example:

```
-> ipv6 ospf interface test admin-state enable
```

10 You can now display the router OSPFv3 settings by using the **show ipv6 ospf** command. The output generated is similar to the following:

```
-> show ipv6 ospf
Status                            = Enabled,
Router ID                         = 30.1.1.2,
# Areas                           = 1,
# Interfaces                      = 3,
Area Border Router                = No,
AS Border Router                  = No,
External Route Tag                = 0,
SPF Hold (seconds)                = 10,
SPF Delay (seconds)               = 5,
MTU checking                      = Enabled,
BFD Status                        = Disabled,
# SPF calculations performed      = 34,
Last SPF run (seconds ago)        = N/A,
# of routes                       = 1,
# of AS external LSAs             = 0,
# of neighbors that are in:
  Full state                      = 1,
  Loading state                   = 0,
  Exchange state                  = 0,
  Exstart state                   = 0,
  2way state                      = 0,
  Init state                      = 0,
  Attempt state                   = 0,
  Down state                      = 0,
Graceful Restart                  = Enabled,
Graceful Restart Status           = Restating,
Graceful Restart Helper           = Enabled,
Graceful Restart Helper Status    = NotHelping
```

11 You can display OSPFv3 area settings using the **show ipv6 ospf area** command. For example:

```
-> show ipv6 ospf area
```

Area ID
As set in Step 7

```
                          Stub   Number of
Area ID          Type     Metric Interfaces
---------------+--------+------+------------------------
0.0.0.0          Normal   NA     2
0.0.0.1          Normal   NA     2
```

12 You can display OSPFv3 interface settings using the **show ipv6 ospf interface** command. For example:

```
-> show ipv6 ospf interface test

Name                              = test
Type                              = BROADCAST,
Admin Status                      = Enabled,
IPv6 Interface Status             = Up,
Oper Status                       = Up,
State                             = DR,
Area                              = 0.0.0.0,
Priority                          = 100,
Cost                              = 1,
Designated Router                 = 3.3.3.3,
Backup Designated Router          = 0.0.0.0,
Hello Interval                    = 1,
Router Dead Interval              = 4,
Retransmit Interval               = 5,
Transit Delay                     = 1,
Ifindex                           = 17,
IPv6 'ifindex'                    = 2071,
MTU                               = 1500,
# of attached neighbors          = 0,
# of state changes                = 0,
Globally reachable prefix #0      = 2071::2/64
```

Area ID
As set in Step 7

13 You can view the contents of the Link-State Database (LDSB) using the **show ipv6 ospf lsdb** command. This command displays the topology information that is provided to/from neighbors. For example:

```
-> show ipv6 ospf lsdb
 Area             Type       Link ID      Advertising Rtr  Sequence #  Age
----------------+----------+------------+-----------------+----------+---------
 0.0.0.0          Router     0            172.28.4.28       8000003b   203
 0.0.0.0          Router     0            172.28.4.29       80000038   35
 0.0.0.0          Network    9            172.28.4.28       80000064   36
 0.0.0.0          Intra AP   16393        172.28.4.28       80000063   36
 0.0.0.0          Inter AP   1            172.28.4.29       80000032   100
 0.0.0.0          Inter AP   2            172.28.4.28       80000032   67
 0.0.0.0          Inter AP   2            172.28.4.29       80000032   100
 0.0.0.0          Inter AP   3            172.28.4.28       80000032   67
 0.0.0.0          Inter AP   3            172.28.4.29       80000033   100
 0.0.0.0          Inter AP   4            172.28.4.29       80000032   73
 0.0.0.0          Link       6            172.28.4.28       80000032   67
 0.0.0.0          Link       7            172.28.4.29       80000033   37
 0.0.0.0          Link       9            172.28.4.28       80000033   75
 0.0.0.3          Router     0            172.28.4.28       80000037   56
 0.0.0.3          Router     0            172.28.4.29       80000038   58
 0.0.0.3          Network    5            172.28.4.29       80000062   122
 0.0.0.3          Intra AP   1            172.28.4.28       80000032   121
 0.0.0.3          Intra AP   1            172.28.4.29       80000032   145
 0.0.0.3          Intra AP   16389        172.28.4.29       80000062   122
 0.0.0.3          Inter AP   1            172.28.4.29       80000032   100
 0.0.0.3          Inter AP   3            172.28.4.29       80000032   100
 0.0.0.3          Inter AP   5            172.28.4.28       80000033   30
 0.0.0.3          Inter AP   6            172.28.4.28       80000032   29
 0.0.0.3          Inter AP   6            172.28.4.29       80000032   22
 0.0.0.3          Inter AP   7            172.28.4.28       80000032   29
 0.0.0.3          Inter AP   7            172.28.4.29       80000032   22
 0.0.0.3          Link       5            172.28.4.29       80000033   145
 0.0.0.3          Link       6            172.28.4.28       80000033   121
```

OSPFv3 Overview

Open Shortest Path First version 3 (OSPFv3) routing is a shortest path first (SPF), or link-state, protocol for IPv6 networks. OSPFv3 is an interior gateway protocol (IGP) that distributes routing information between routers in a Single Autonomous System (AS). OSPFv3 chooses the least-cost path as the best path.

Each participating router distributes its local state (i.e., the router's usable interfaces, local networks, and reachable neighbors) throughout the AS by flooding Link-State Advertisements (LSAs). Each router maintains a link-state database (LSDB) describing the entire topology. The LSDB is built from the collected LSAs of all routers within the AS. Each multi-access network that has at least two attached routers has a designated router and a backup designated router. The designated router floods an LSA for the multi-access network.

When a router starts, it uses the OSPFv3 Hello Protocol to discover neighbors and elect a designated router for the network. Neighbors are dynamically detected by sending Hello packets to a multicast address. The router sends Hello packets to its neighbors and in turn receives their Hello packets.

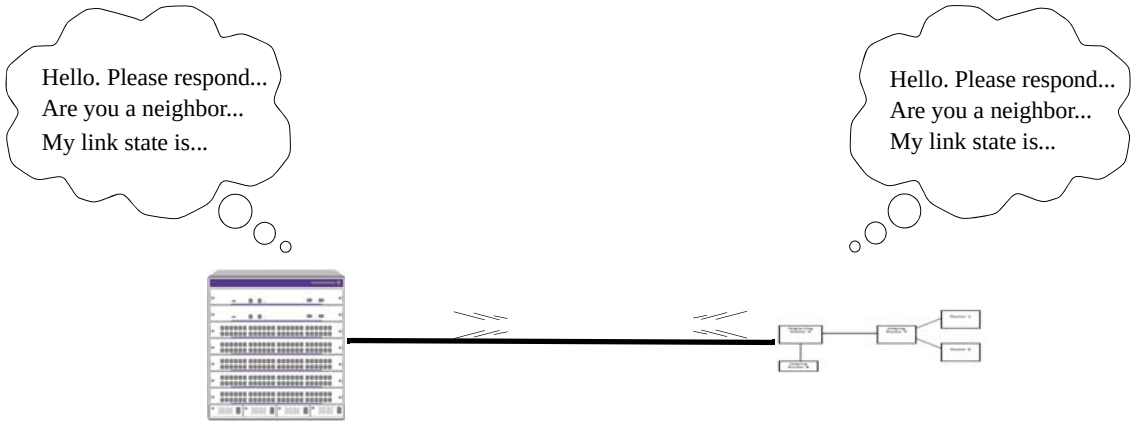


Figure 2-1 :OSPFv3 Hello Protocol

The router will attempt to form full adjacencies with all of its newly acquired neighbors. Only some pairs, however, will be successful in forming full adjacencies. Topological databases are synchronized between pairs of fully adjacent routers.

Adjacencies control the distribution of routing protocol packets. Routing protocol packets are sent and received only on adjacencies. In particular, distribution of topological database updates proceeds along adjacencies.

Link state is also advertised when a router's state changes. A router's adjacencies are reflected in the contents of its link state advertisements. This relationship between adjacencies and link state allows the protocol to detect downed routers in a timely fashion.

AS link state advertisements are flooded throughout the AS, across areas. Area link state advertisements are flooded to routers within the same area. The flooding algorithm ensures that all routers within a given area have exactly the same LSDB. This database consists of the collection of link state advertisements received from each router belonging to the area. From this database each router calculates a shortest-path tree. This shortest-path tree in turn yields a routing table for the protocol.

OSPFv3 Areas

OSPFv3 allows collections of contiguous networks and hosts to be grouped together as an *area*. Each area runs a separate copy of the basic link-state routing algorithm (usually called SPF). This means that each area has its own topological database, as explained in the previous section.

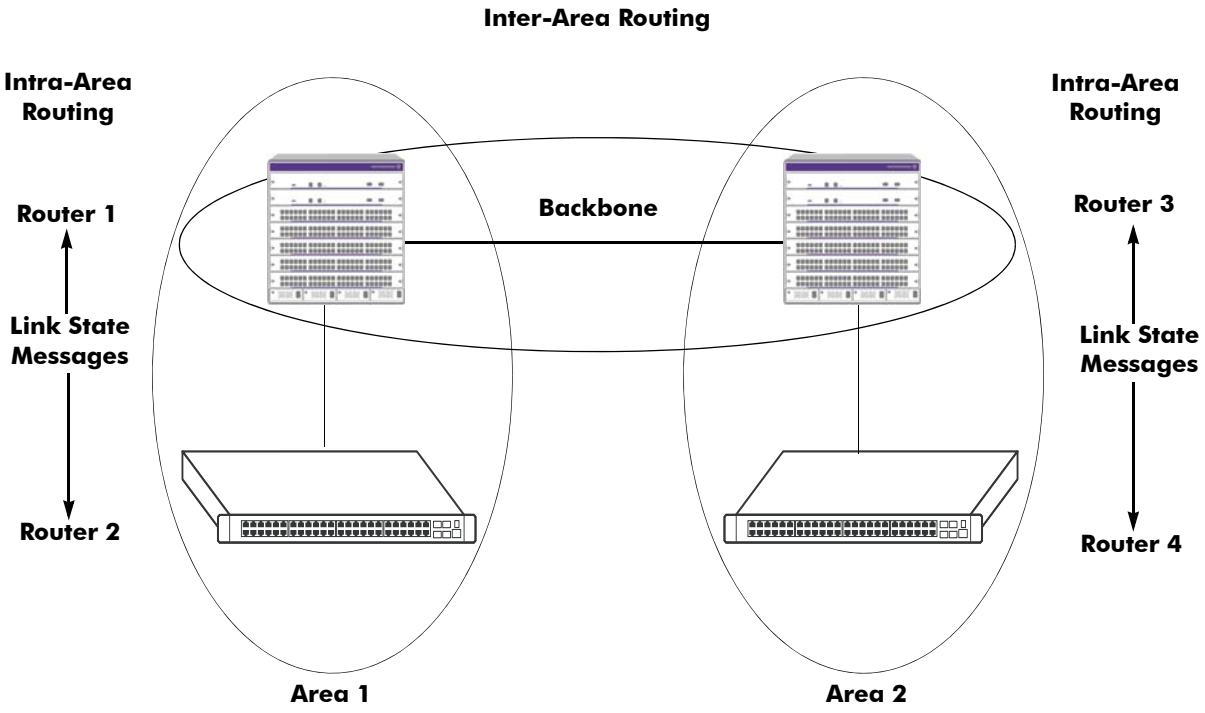


Figure 2-2 : OSPFv3 Intra-Area and Inter-Area Routing

An area's topology is visible only to the members of the area. Conversely, routers internal to a given area know nothing of the detailed topology external to the area. This isolation of knowledge enables the protocol to reduce routing traffic by concentrating on small areas of an AS, as compared to treating the entire AS as a single link-state domain.

Each router that participates in a specific area maintains an LSDB containing topological information for that area. If the router participates in multiple areas, then it will maintain a separate database for each area to which the router belongs. LSAs are flooded throughout an area to ensure that all participating routers have an identical LSDB for that area.

A router connected to multiple areas is identified as an *area border router* (ABR). All ABRs must also belong to a *backbone* area (also known as area 0). The backbone is responsible for distributing routing information between areas. Although the backbone is an area itself, it consists of area border routers and must also have links to all areas to which it will transfer information.The topology of the backbone area is invisible to each of the areas, while the backbone itself knows nothing of the topology of the areas.

All routers in an area must agree on that area's parameters. Since a separate copy of the link-state algorithm is run in each area, most configuration parameters are defined on a per-router basis. All routers belonging to an area must agree on that area's configuration. Misconfiguration will keep neighbors from forming adjacencies between themselves, and OSPFv3 will not function.

Classification of Routers

When an AS is split into OSPFv3 areas, the routers are further divided according to function into the following four overlapping categories:

- **Internal area router.** A router with all directly connected networks belonging to the same area. Each internal router shares the same LSDB with other routers within the same area.
- **Area border router (ABR)**. A router that attaches to multiple areas and to the backbone area. ABRs maintain a separate LSDB for each area to which it is connected, in addition to an AS and link-local database. The topological information from each area LSDB is condensed by the ABR and flooded to other areas.
- **Designated router (DR).** An elected router that is responsible for generating LSAs and maintaining the LSDB for the subnet to which the router is connected. The DR updates the LSDB by exchanging database updates with adjacent, non-designated routers on the network.
- **AS boundary router**. A router that exchanges routing information with routers belonging to other Autonomous Systems. Such a router may also advertise external routes throughout the Autonomous System. The path to each AS boundary router is known by every router in the AS. This classification is completely independent of the previous classifications (i.e., internal and area border routers). AS boundary routers may be internal or area border routers.

Virtual Links

It is possible to define areas in such a way that the backbone is no longer contiguous. (This is not an ideal OSPFv3 configuration, and maximum effort should be made to avoid this situation.) In this case the system administrator must restore backbone connectivity by configuring *virtual links*.

Virtual links can be configured between any two backbone routers that have a connection to a common non-backbone area. The protocol treats two routers joined by a virtual link as if they were connected by an unnumbered point-to-point network. The routing protocol traffic that flows along the virtual link uses intra-area routing only, and the physical connection between the two routers is not managed by the network administrator (i.e., there is no dedicated connection between the routers as there is with the OSPFv3 backbone).

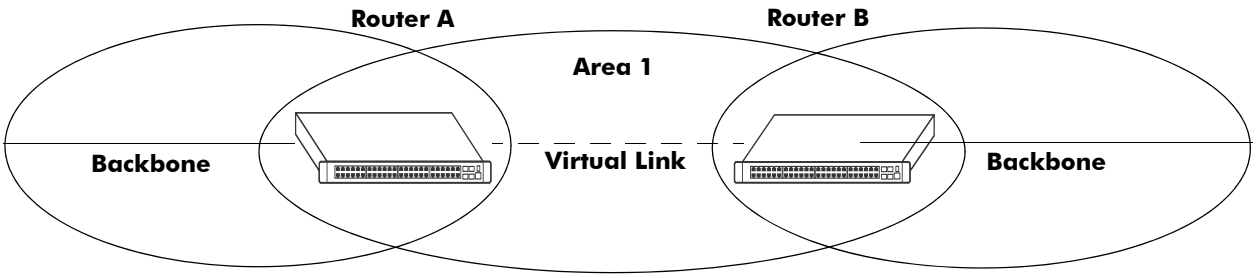


Figure 2-3 : OSPFv3 Routers Connected with a Virtual Link

In the above diagram, Router A and Router B are connected via a virtual link in Area 1, which is known as a transit area. See “Creating Virtual Links” on page 2-21 for more information.

Stub Areas

OSPFv3 allows certain areas to be configured as *stub areas*. A stub area is an area with routers that have no AS external Link State Advertisements (LSAs).

In order to take advantage of the OSPFv3 stub area support, default routing must be used in the stub area. This is accomplished by configuring one or more of the stub area's border routers to advertise a default route into the stub area. The default routes will match any destination that is not explicitly reachable by an intra-area or inter-area path (i.e., AS external destinations).

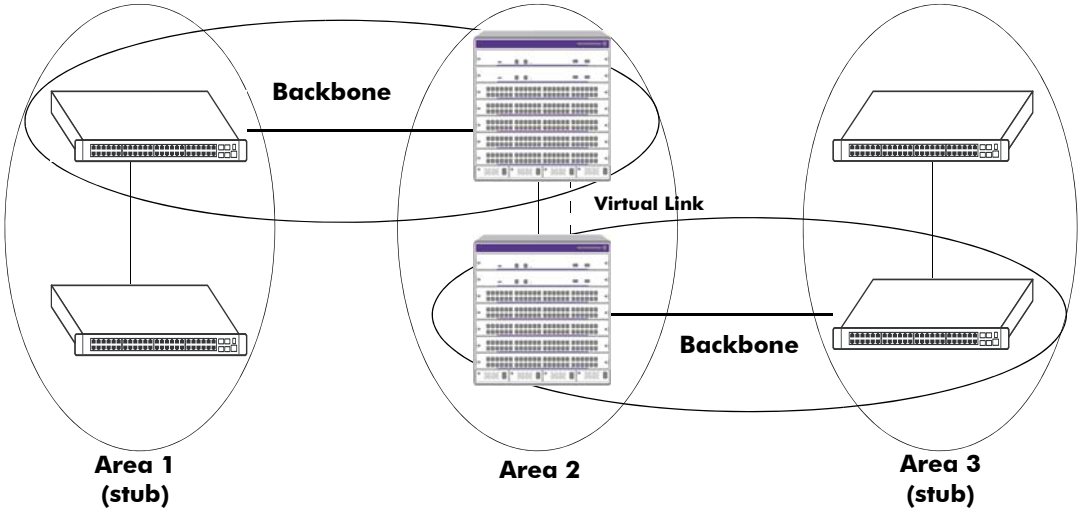


Figure 2-4 : OSPFv3 Stub Area

Area 1 and Area 3 could be configured as stub areas. Stub areas are configured using the OSPFv3 **ipv6 ospf area** command, described in "Creating an Area" on page 2-16. For more overview information on areas, see "OSPFv3 Areas" on page 2-8.

The OSPFv3 protocol ensures that all routers belonging to an area agree on whether the area has been configured as a stub. This guarantees that no confusion will arise in the flooding of AS external advertisements.

Two restrictions on the use of stub areas are:

- Virtual links cannot be configured through stub areas.
- AS boundary routers cannot be placed internal to stub areas.

Not-So-Stubby-Areas

NSSA (not-so-stubby area), is an extension of the stub area. NSSA is similar to a stub area in many ways: AS-external LSAs are not flooded into an NSSA and virtual links are not allowed in an NSSA. The primary difference is that selected external routing information can be imported into an NSSA and then redistributed into the rest of the OSPF routing domain. These routes are imported into the NSSA using a new LSA type: Type-7 LSA. Type-7 LSAs are flooded within the NSSA and are translated at the NSSA boundary into AS-external LSAs so as to convey the external routing information to other areas.

NSSAs enable routers with limited resources to participate in OSPF routing while also allowing the import of a selected number of external routes into the area. For example, an area which connects to a small external routing domain running RIP may be configured as an NSSA. This will allow the import of RIP routes into this area and the rest of the OSPF routing domain and at the same time, prevent the flooding of other external routing information (learned, for example, through RIP) into this area.

All routers in an NSSA must have their OSPF area defined as an NSSA. To configure otherwise will ensure that the router will be unsuccessful in establishing an adjacency in the OSPF domain.

NSSAs are configured using the OSPFv3 **ipv6 ospf area** command, described in “Creating an Area” on page 2-16. For more overview information on areas, see “OSPFv3 Areas” on page 2-8.

NSSA related configurations are described in “Configuring OSPFv3 NSSA Parameters” on page 2-18.

Equal Cost Multi-Path (ECMP) Routing

Using information from its continuously updated databases, OSPFv3 calculates the shortest path to a given destination. Shortest path is determined from metric values at each hop along a path. At times, two or more paths to the same destination will have the same metric cost.

In the network illustration below, there are two paths from Source router A to Destination router B. One path traverses two hops at routers X and Y and the second path traverses two hops at M and N. If the total cost through X and Y to B is the same as the cost via M and N to B, then these two paths have equal cost. In this version of OSPFv3 both paths will be stored and used to transmit data.

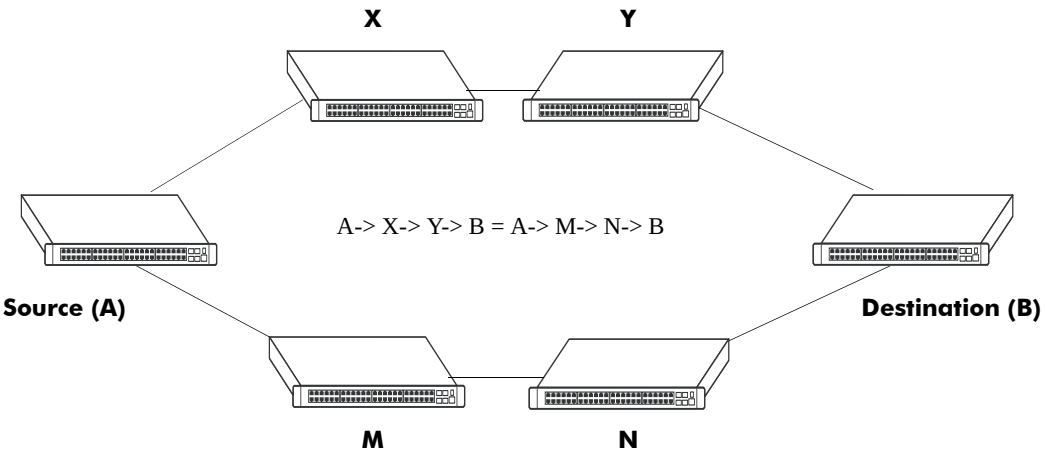


Figure 2-5 : Multiple Equal Cost Paths

Delivery of packets along equal paths is based on flows rather than a round-robin scheme. Equal cost is determined based on standard routing metrics. However, other variables, such as line speed, are not considered. So it is possible for OSPFv3 to decide two paths have an equal cost even though one may contain faster links than another.

Non Broadcast OSPF Routing

OSPFv3 can operate in two modes on non-broadcast networks: NBMA and point-to-multipoint. The interface type for the corresponding network segment should be set to non-broadcast or point-to-multipoint, respectively.

For non-broadcast networks, neighbors should be statically configured. For NBMA neighbors, the eligibility option must be enabled for the neighboring router to participate in Designated Router (DR) election.

For the correct working of an OSPFv3 NBMA network, a fully meshed network is mandatory. Also, the neighbor eligibility configuration for a router on every other router should match the routers interface priority configuration.

See “Configuring Static Neighbors” on page 2-28 for more information and setting up static neighbors.

Graceful Restart on Switches with Redundant CMMs

A chassis-based switch with two Chassis Management Modules (CMMs) can support redundancy where if for any reason the primary CMM fails or goes offline, the secondary CMM is instantly notified. The secondary CMM automatically assumes the primary role. This switch between the primary and secondary CMMs is known as *takeover*.

When a takeover occurs, which can be planned (e.g., the user performs the takeover) or unplanned (e.g., the primary CMM unexpectedly fails), an OSPFv3 router must reestablish full adjacencies with all its previously fully adjacent neighbors. This time period between the restart and the reestablishment of adjacencies is termed *graceful restart*.

The OSPFv3 router attempting a graceful restart originates link-local Opaque-LSAs, called Grace-LSAs, announcing the intention to perform a graceful restart and requests for a grace period. On receiving Grace-LSA, the restart-aware neighboring OSPFv3 routers detect that the neighboring router is performing a graceful restart and commence executing in ‘helper mode’. During the grace period, these neighbors continue to announce the 'restarting' router in their LSAs as if it were fully adjacent provided the grace period has not expired, graceful restart helper functionality is enabled, and network topology remains stable.

The restarting router then re-establishes adjacencies with all its neighboring routers, updating its link-state database with link-state received from these helper neighbors. Once all adjacencies are established, the restarting router flushes its grace LSAs signaling the successful termination of graceful restart and restoration of normal OSPFv3 operation.

In the network illustration below, when router X gracefully restarts, it sends out a Grace-LSA on all the active OSPFv3 interfaces. When the neighboring routers (A and Y) receive the Grace-LSA, they go into helping mode and help to restart the router by not bringing down the adjacency (helping routers advertise the restarting router as a fully adjacent neighbor in their LSAs irrespective of their current adjacency state) and help to quickly rebuild the link state database.

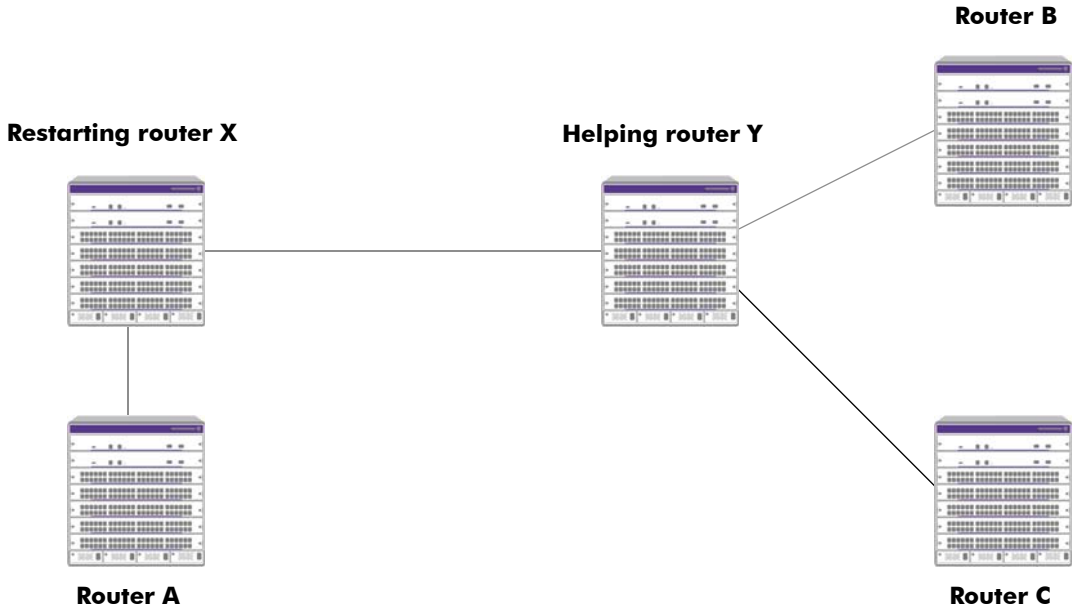


Figure 2-6 : OSPFv3 Graceful Restart Helping and Restarting Router Example

See “Configuring Redundant CMMs for Graceful Restart” on page 2-29 for more information on configuring graceful restart.

Configuring OSPFv3

Configuring OSPFv3 on a router requires several steps. Depending on your requirements, you may not need to perform all of the steps listed below.

By default, OSPFv3 is enabled on the router. Configuring OSPFv3 consists of these tasks:

- Set up the basics of the OSPFv3 network by configuring the required VLANs, assigning ports to the VLANs, and assigning router identification numbers to the routers involved. This is described in "Preparing the Network for OSPFv3" on page 2-15.
- Load OSPFv3. When the image file for advanced routing is installed, you must load the OSPFv3 code. The commands for loading OSPFv3 are described in "Activating OSPFv3" on page 2-15.
- Create any desired OSPFv3 areas, including the backbone area if one is required. Note that a backbone area is not necessary if there is only one area. The commands to create areas and backbone areas are described in "Creating an OSPFv3 Area" on page 2-16. OSPFv3 will run with the default area parameters, but different networks may benefit from modifying the parameters.
- Create OSPFv3 interfaces. OSPFv3 interfaces are created and assigned to areas. Creating interfaces and assigning interfaces is described in "Creating OSPFv3 Interfaces" on page 2-19.
- Set interface parameters (optional). OSPFv3 will run with the default interface parameters, but different networks may benefit from modifying the parameters. Modifying interface parameters is described in "Modifying Interface Parameters" on page 2-19
- Configure virtual links (optional). A virtual link is used to establish backbone connectivity when two backbone routers are not physically contiguous. To create a virtual link, see "Creating Virtual Links" on page 2-21.
- Configure redistribution using route maps (optional). Redistribution allows the control of how routes are advertised into the OSPFv3 network from outside the Autonomous System. Configuring redistribution is described in "Configuring Redistribution" on page 2-21.
- Create static neighbors (optional). These commands allow you to statically configure neighbors. See "Configuring Static Neighbors" on page 2-28.
- Configure router capabilities (optional). There are several commands that influence router operation. These are covered briefly in a table in "Configuring Router Capabilities" on page 2-27.
- Configure redundant switches for graceful OSPFv3 restart (optional). Configuring switches with redundant switches for graceful restart is described in "Configuring Redundant CMMs for Graceful Restart" on page 2-29.

At the end of the chapter is a simple OSPFv3 network diagram with instructions on how it was created on a router-by-router basis. See "OSPFv3 Application Example" on page 2-30 for more information.

Preparing the Network for OSPFv3

OSPFv3 operates on top of normal switch functions, using existing ports, virtual ports, VLANs, etc. The following network components should already be configured:

- **Configure VLANs that are to be used in the OSPFv3 network**. VLANS should be created for interfaces that will participate in the OSPFv3 network. VLAN configuration is described in "Configuring VLANs" in the *OmniSwitch AOS Release 8 Network Configuration Guide*.
- **Assign IPv6 interfaces to the VLANs.** IPv6 interfaces must be assigned to the VLAN. Assigning IPv6 interfaces is described in "Configuring IP" in the *OmniSwitch AOS Release 8 Network Configuration Guide*.
- **Assign ports to the VLANs**. The physical ports participating in the OSPFv3 network must be assigned to the created VLANs. Assigning ports to a VLAN is described in "Assigning Ports to VLANs" in the *OmniSwitch AOS Release 8 Network Configuration Guide*.
- **Set the router identification number**. (optional) The routers participating in the OSPFv3 network must be assigned a router identification number. This number is specified using the standard dotted decimal format (e.g., 1.1.1.1) but may not consist of all zeros (0.0.0.0). Router identification number assignment is discussed in "Configuring IP" in the *OmniSwitch AOS Release 8 Network Configuration Guide*. If this is not done, the router identification number is automatically the primary interface address.

Activating OSPFv3

To run OSPFv3 on the router, the advanced routing image must be installed. See the *OmniSwitch AOS Release 8 Switch Management Guide* for information on how to install image files.

After the image file has been installed onto the router, you will need to load the OSPFv3 software into memory as described below:

Loading the Software

To load the OSPFv3 software into the router's running configuration, enter the **ipv6 load ospf** command at the system prompt:

```
-> ipv6 load OSPF
```

The OPSFv3 software is now loaded into memory.

Configuring the OSPFv3 Administrative Status

When the OSPFv3 software is loaded into the router's running configuration (either through the CLI or on startup), it is administratively enabled by default. To change the OSPFv3 administrative status, use the **ipv6 ospf admin-state** command. For example, the following commands disable and enable OSPFv3 on the router:

```
-> ipv6 ospf admin-state disable
-> ipv6 ospf admin-state enable
```

Removing OSPFv3 from Memory

To remove OSPFv3 from the router memory, it is necessary to manually edit the **boot.cfg** file. The **boot.cfg** file is an ASCII text-based file that controls many of the switch parameters. Open the file and delete all references to OSPFv3.

For the operation to take effect the switch needs to be rebooted.

Creating an OSPFv3 Area

OSPFv3 allows a set of network devices in an Autonomous System (AS) to be grouped together in *areas*.

There can be more than one router in an area. Likewise, there can be more than one area on a single router (in effect, making the router the Area Border Router (ABR) for the areas involved), but standard networking design does not recommended that more than three areas be handled on a single router.

Note that configuring a backbone area for a router is required if the router is going to participate in more than one area.

Areas are named using 32-bit dotted decimal format (e.g., 1.1.1.1). Area 0.0.0.0 is reserved for the backbone.

Creating an Area

To create an area and associate it with a router, enter the **ipv6 ospf area** command with the area identification number at the CLI prompt, as shown:

```
-> ipv6 ospf area 1.1.1.1
```

Area 1.1.1.1 will now be created on the router with the default parameters.

The backbone is always area 0.0.0.0. To create this area on a router, you would use the above command, but specify the backbone, as shown:

```
-> ipv6 ospf area 0.0.0.0
```

The backbone would now be attached to the router, making it an Area Border Router (ABR).

Specifying an Area Type

When creating areas, an area type can be specified (normal, stub, or NSSA). Area types are described above in “OSPFv3 Areas” on page 2-8. To specify an area type, use the **ipv6 ospf area** command as shown:

```
-> ipv6 ospf area 1.1.1.1 type stub
```

Note. By default, an area is a normal area. The type keyword would be used to change a normal area into stub or NSSA.

See “Configuring OSPFv3 NSSA Parameters” on page 2-18 for more information on configuring OSPFv3 NSSA area parameters.

Configuring Route Summarization Range

Summarize option can be used to summarize many area routes into a single advertisement at an area boundary. Prefixes that fall within the summary range can be shared with another area through the summary prefix.

To configure the inter-area route summarization, use the **summarize** option in **ipv6 ospf area** command:

```
-> ipv6 ospf area 0.0.0.1 summarize 2001:1::/64
```

To filter the routes specified in the summarization range, use the **filter** option. These filtered routes are not advertised into another area.

```
-> ipv6 ospf area 0.0.0.1 summarize 2001:1::/64 filter
```

Configuring Area Default Metrics

The default metric configures the metric that an area border router (ABR) will advertise into the stub or NSSA. Use the **ipv6 ospf area** command to modify the default metric for a stub or NSSA area.

```
-> ipv6 ospf area 1.1.1.1 type stub default-metric 10
```

Enabling and Disabling Summarization for a Stub or NSSA

To configure whether or not summary routes are imported into the stub or NSSA as Type-3 summary-LSAs, use the **ipv6 ospf area area-summary** command.

When set to **noareasummary** option, inter-area LSAs will neither originate or propagate into the stub or NSSA. Only a default route will be advertised into the stub or NSSA.

```
-> ipv6 ospf area 1.1.1.1 area-summary noareasummary
```

When set to **sendareasummary** option, inter-area LSAs will be summarized and propagated into the stub or NSSA.

```
-> ipv6 ospf area 1.1.1.1 area-summary sendareasummary
```

Displaying Area Status

You can check the status of the newly created area by using the **show** command, as demonstrated:

```
-> show ipv6 ospf area 1.1.1.1
```

or

```
-> show ipv6 ospf area
```

The first example gives specifics about area 1.1.1.1, and the second example shows all areas configured on the router.

To display the parameters of an area, use the **show ipv6 ospf area** command as follows:

```
-> show ipv6 ospf area 1.1.1.1
```

Deleting an Area

To delete an area, enter the **ipv6 ospf area** command as shown:

```
-> no ipv6 ospf area 1.1.1.1
```

Configuring OSPFv3 NSSA Parameters

The following NSSA parameters can be configured.

Configure NSSA Translator Role

To configure whether or not an NSSA border router must unconditionally translate Type-7 LSAs into Type-5 LSAs, use the **ipv6 ospf area nssa-translator-role** command.

When set to **always** option, NSSA border router always translates Type-7 LSAs into Type-5 LSAs regardless of the translator state of other NSSA border routers. When set to **candidate** option, NSSA border router participates in the translator election process.

```
-> ipv6 ospf area 0.0.0.1 nssa-translator role always
-> ipv6 ospf area 0.0.0.1 nssa-translator role candidate
```

Configure NSSA Translator Stability Interval

To configure NSSA stability interval, use the **ipv6 ospf area nssa-translator-stab-interval** command. Stability interval is the duration for which a Type-7 translator will continue in the translator role after another NSSA border router translator has assumed the role. The default stability interval is 40 seconds. For example,

```
-> ipv6 ospf area 0.0.0.1 nssa-translator-stab-interval 60
```

Configure NSSA Summarization

To configure an NSSA summary of IPv6 prefix in the given area, use **ipv6 ospf area nssa-summarize** command. Any NSSA LSAs within the area subsumed by IPv6 prefix will be summarized into other areas as an external LSA with a prefix of IPv6 prefix. Use **filter** option to suppress the external LSA.

For example,

```
-> ipv6 ospf area 2 nssa-summarize c000::/64
-> ipv6 ospf area 2 nssa-summarize c000::/64 filter
```

Creating OSPFv3 Interfaces

Once areas have been established, interfaces need to be created and assigned to the areas. (Creating areas is described in "Creating an Area" on page 2-16 above.)

To create an interface and assign it to an area, enter the **ipv6 ospf interface area** command with an interface name and an area identification number, as shown:

```
-> ipv6 ospf interface vlan-213 area 1.1.1.1
```

Note. The interface name *cannot* have spaces.

The interface can be deleted by using the **no** keyword, as shown:

```
-> no ipv6 ospf interface vlan-213
```

An interface can be removed from an area by reassigning it to a new area.

Once an interface has been created, you can check its status and configuration by using the **show ipv6 ospf interface** command, as demonstrated:

```
-> show ipv6 ospf interface vlan-213
```

Instructions for interface parameter options are described in "Modifying Interface Parameters" on page 2-19.

Configuring the Interface Administrative Status

When an OSPFv3 interface is created and assigned an area, it is administratively enabled by default. To change the administrative status of the interface, use the **ipv6 ospf interface admin-state** command with the interface IP address or interface name, as shown:

```
-> ipv6 ospf interface vlan-213 admin-state disable
-> ipv6 ospf interface vlan-213 admin-state enable
```

Configuring the Loopback0 Interface

Unlike with OSPFv2, the OSPFv3 Loopback0 interface is not automatically advertised to its neighbor. To advertise the Loopback0 interface, configure it as a point-to-point interface. For example:

```
-> ipv6 interface "ipv6-loopback0" loopback0
-> ipv6 address 2001:6b0:17:ff00:ffff:0:b:b1/128 "ipv6-loopback0"
-> ipv6 ospf interface "ipv6-loopback0" area 0.0.0.0
-> ipv6 ospf interface ipv6-loopback0 area 0.0.0.0 type point-to-point
```

Modifying Interface Parameters

There are several interface parameters that can be modified on a specified interface. Most of these deal with timer settings.

The cost parameter and the priority parameter help to determine the cost of the route using this interface, and the chance that this interface's router will become the designated router, respectively.

The following table shows the various interface parameters that can be set:

ipv6 ospf interface dead-interval	Configures the OSPFv3 interface dead interval. If no hello packets are received in this interval from a neighboring router, the neighbor is considered dead.
ipv6 ospf interface hello-interval	Configures the OSPFv3 hello interval.
ipv6 ospf interface cost	Configures the OSPFv3 interface cost. A cost metric refers to the network path preference assigned to certain types of traffic.
ipv6 ospf interface priority	Configures the OSPFv3 interface priority. The priority number helps determine if this router will become the designated router.
ipv6 ospf interface retrans-interval	Configures the OSPFv3 interface retransmit interval. The number of seconds between link state advertisement retransmissions for adjacencies belonging to this interface.
ipv6 ospf interface transit-delay	Configures the OSPFv3 interface transit delay. The estimated number of seconds required to transmit a link state update over this interface.

These parameters can be added any time. (See "Creating OSPFv3 Interfaces" on page 2-19 for more information.) For example, to set the dead interval to 50 and the cost to 100 on interface vlan-213, enter the following:

```
-> ipv6 ospf interface vlan-213 dead-interval 50
-> ipv6 ospf interface vlan-213 cost 100
```

To set the priority to 100, and the retransmit interval to 10 on interface vlan-213, enter the following:

```
-> ipv6 ospf interface vlan-213 priority 100 retrans-interval 10
```

To set the hello interval to 5000 on interface vlan-213, enter the following:

```
-> ipv6 ospf interface vlan-213 hello-interval 5000
```

Interface Types

Different interface types can be configured based on the network type connected. OmniSwitch supports point-to-point, point-to-multipoint, non-broadcast multiple access (NBMA), and broadcast interface types.

To configure the interface type, use the **ipv6 ospf interface type** command. For example, to set the interface type as point-to-multipoint for the interface vlan-213, enter the following:

```
-> ipv6 ospf interface vlan-213 type point-to-multipoint
```

Creating Virtual Links

To create a virtual link, commands must be submitted to the routers at both ends of the link. The router being configured should point to the other end of the link, and both routers must have a common area.

When entering the **ipv6 ospf virtual-link** command, it is necessary to enter the Router ID of the far end of the link, and the area ID that both ends of the link share.

For example, a virtual link needs to be created between Router A (router ID 1.1.1.1) and Router B (router ID 2.2.2.2). We must:

1 Establish a transit area between the two routers using the commands discussed in "Creating an OSPFv3 Area" on page 2-16 (in this example, we will use Area 0.0.0.1).

2 Then use the **ipv6 ospf virtual-link** command on Router A as shown:

```
-> ipv6 ospf virtual-link area 0.0.0.1 router 2.2.2.2
```

3 Next, enter the following command on Router B:

```
-> ipv6 ospf virtual-link area 0.0.0.1 router 1.1.1.1
```

Now there is a virtual link across Area 0.0.0.1 linking Router A and Router B.

4 To display virtual links configured on a router, enter the following **show** command:

```
-> show ipv6 ospf virtual-link
```

5 To delete a virtual link, enter the **ipv6 ospf virtual-link** command with the area and far end router information, as shown:

```
-> no ipv6 ospf virtual-link area 0.0.0.1 router 2.2.2.2
```

Modifying Virtual Link Parameters

There are several parameters for a virtual link (such as hello-interval and dead-interval that can be modified at the time of the link creation. They are described in the **ipv6 ospf virtual-link** command description. These parameters are identical in function to their counterparts in the section "Modifying Interface Parameters" on page 2-19.

Configuring Redistribution

It is possible to learn and advertise IPv6 routes between different protocols. Such a process is referred to as route redistribution and is configured using the **ipv6 redist** command.

Redistribution uses route maps to control how external routes are learned and distributed. A route map consists of one or more user-defined statements that can determine which routes are allowed or denied access to the network. In addition a route map may also contain statements that modify route parameters before they are redistributed.

When a route map is created, it is given a name to identify the group of statements that it represents. This name is required by the **ipv6 redist** command. Therefore, configuring route redistribution involves the following steps:

1 Create a route map, as described in "Using Route Maps" on page 2-22.

2 Configure redistribution to apply a route map, as described in "Configuring Route Map Redistribution" on page 2-25.

Note. An OSPFv3 router automatically becomes an Autonomous System Border Router (ASBR) when redistribution is configured on the router.

Using Route Maps

A route map specifies the criteria that are used to control redistribution of routes between protocols. Such criteria is defined by configuring route map statements. There are three different types of statements:

- **Action**. An action statement configures the route map name, sequence number, and whether or not redistribution is permitted or denied based on route map criteria.
- **Match.** A match statement specifies criteria that a route must match. When a match occurs, then the action statement is applied to the route.
- **Set.** A set statement is used to modify route information before the route is redistributed into the receiving protocol. This statement is only applied if all the criteria of the route map is met and the action permits redistribution.

The **ip route-map** command is used to configure route map statements and provides the following **action**, **match**, and **set** parameters:

ip route-map action ...	ip route-map match ...	ip route-map set ...
permit **deny**	**ip-address** **ip-nexthop** **ipv6-address** **ipv6-nexthop** **tag** **ipv4-interface** **ipv6-interface** **metric** **route-type**	**metric** **metric-type** **tag** **community** **local-preference** **level** **ip-nexthop** **ipv6-nexthop**

Refer to the “IP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the **ip route-map** command parameters and usage guidelines.

Once a route map is created, it is then applied using the **ipv6 redist** command. See “Configuring Route Map Redistribution” on page 2-25 for more information.

Creating a Route Map

When a route map is created, it is given a name (up to 20 characters), a sequence number, and an action (permit or deny). Specifying a sequence number is optional. If a value is not configured, then the number 50 is used by default.

To create a route map, use the **ip route-map** command with the **action** parameter. For example,

```
-> ip route-map ospf-to-rip sequence-number 10 action permit
```

The above command creates the ospf-to-rip route map, assigns a **sequence number** of 10 to the route map, and specifies a **permit** action.

To optionally filter routes before redistribution, use the **ip route-map** command with a **match** parameter to configure match criteria for incoming routes. For example,

```
-> ip route-map ospf-to-rip sequence-number 10 match tag 8
```

The above command configures a match statement for the ospf-to-rip route map to filter routes based on their tag value. When this route map is applied, only OSPFv3 routes with a tag value of eight are redistributed into the RIPng network. All other routes with a different tag value are dropped.

Note. Configuring match statements is not required. However, if a route map does not contain any match statements and the route map is applied using the **ipv6 redist** command, the router redistributes *all* routes into the network of the receiving protocol.T

To modify route information before it is redistributed, use the **ip route-map** command with a **set** parameter. For example,

```
-> ip route-map ospf-to-rip sequence-number 10 set tag 5
```

The above command configures a set statement for the ospf-to-rip route map that changes the route tag value to five. Because this statement is part of the ospf-to-rip route map, it is only applied to routes that have an existing tag value equal to eight.

The following is a summary of the commands used in the above examples:

```
-> ip route-map ospf-to-rip sequence-number 10 action permit
-> ip route-map ospf-to-rip sequence-number 10 match tag 8
-> ip route-map ospf-to-rip sequence-number 10 set tag 5
```

To verify a route map configuration, use the **show ip route-map** command:

```
-> show ip route-map
Route Maps: configured: 1 max: 200
Route Map: ospf-to-rip Sequence Number: 10 Action permit
  match tag 8
  set tag 5
```

Deleting a Route Map

Use the **no** form of the **ip route-map** command to delete an entire route map, a route map sequence, or a specific statement within a sequence.

To delete an entire route map, enter **no ip route-map** followed by the route map name. For example, the following command deletes the entire route map named redistipv6:

```
-> no ip route-map redistipv6
```

To delete a specific sequence number within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the actual number. For example, the following command deletes sequence 10 from the redistipv6 route map:

```
-> no ip route-map redistipv6 sequence-number 10
```

Note that in the above example, the redistripv6 route map is not deleted. Only those statements associated with sequence 10 are removed from the route map.

To delete a specific statement within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the sequence number for the statement, then either **match** or **set** and the match or set parameter and value. For example, the following command deletes only the match tag 8 statement from route map redistipv6 sequence 10:

```
-> no ip route-map redistipv6 sequence-number 10 match tag 8
```

Configuring Route Map Sequences

A route map may consist of one or more sequences of statements. The sequence number determines which statements belong to which sequence and the order in which sequences for the same route map are processed.

To add match and set statements to an existing route map sequence, specify the same route map name and sequence number for each statement. For example, the following series of commands creates route map rm_1 and configures match and set statements for the rm_1 sequence 10:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 set metric 1
```

To configure a new sequence of statements for an existing route map, specify the same route map name but use a different sequence number. For example, the following command creates a new sequence 20 for the rm_1 route map:

```
-> ip route-map rm_1 sequence-number 20 action permit
-> ip route-map rm_1 sequence-number 20 match ipv6-interface to-finance
-> ip route-map rm_1 sequence-number 20 set metric 5
```

The resulting route map appears as follows:

```
-> show ip route-map rm_1
Route Map: rm_1 Sequence Number: 10 Action permit
  match tag 8
  set metric 1
Route Map: rm_1 Sequence Number: 20 Action permit
  match ip6 interface to-finance
  set metric 5
```

Sequence 10 and sequence 20 are both linked to route map rm_1 and are processed in ascending order according to their sequence number value. Note that there is an implied logical OR between sequences. As a result, if there is no match for the tag value in sequence 10, then the match interface statement in sequence 20 is processed. However, if a route matches the tag 8 value, then sequence 20 is not used. The set statement for whichever sequence was matched is applied.

A route map sequence may contain multiple match statements. If these statements are of the same kind (e.g., match tag 5, match tag 8, etc.) then a logical OR is implied between each like statement. If the match statements specify different types of matches (e.g. match tag 5, match ip4 interface to-finance, etc.), then a logical AND is implied between each statement. For example, the following route map sequence will redistribute a route if its tag is either 8 or 5:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
```

The following route map sequence will redistribute a route if the route has a tag of 8 or 5 *and* the route was learned on the IPv4 interface to-finance:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 match ipv4-interface to-finance
```

Configuring Access Lists

An IP access list provides a convenient way to add multiple IPv4 or IPv6 addresses to a route map. Using an access list avoids having to enter a separate route map statement for each individual IP address. Instead, a single statement is used that specifies the access list name. The route map is then applied to all the addresses contained within the access list.

Configuring an IP access list involves two steps: creating the access list and adding IP addresses to the list. To create an IP access list, use the **ip access-list** command (IPv4) or the **ipv6 access-list** command (IPv6) and specify a name to associate with the list. For example,

```
-> ip access-list ipaddr
-> ipv6 access-list ip6addr
```

To add addresses to an access list, use the **ip access-list address** (IPv4) or the **ipv6 access-list address** (IPv6) command. For example, the following commands add addresses to an existing access list:

```
-> ip access-list ipaddr address 16.24.2.1/16
-> ipv6 access-list ip6addr address 2001::1/64
```

Use the same access list name each time the above commands are used to add additional addresses to the same access list. In addition, both commands provide the ability to configure if an address and/or its matching subnet routes are permitted (the default) or denied redistribution. For example:

```
-> ip access-list ipaddr address 16.24.2.1/16 action deny redist-control all-
subnets
-> ipv6 access-list ip6addr address 2001::1/64 action permit redist-control no-
subnets
```

For more information about configuring access list commands, see the “IP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Route Map Redistribution

The **ipv6 redist** command is used to configure the redistribution of routes from a source protocol into the OSPFv3 destination protocol. This command is used on the OSPFv3 router that will perform the redistribution.

Note. A router automatically becomes an Autonomous System Border Router (ASBR) when redistribution is configured on the router.

A source protocol is a protocol from which the routes are learned. A destination protocol is the one into which the routes are redistributed. Make sure that both protocols are loaded and enabled before configuring redistribution.

Redistribution applies criteria specified in a route map to routes received from the source protocol. Therefore, configuring redistribution requires an existing route map. For example, the following command configures the redistribution of OSPFv3 routes into the RIPng network using the ospf-to-rip route map:

```
-> ipv6 redist ospf into rip route-map ospf-to-rip
```

OSPFv3 routes received by the router interface are processed based on the contents of the ospf-to-rip route map. Routes that match criteria specified in this route map are either allowed or denied redistribution into the RIPng network. The route map may also specify the modification of route information before the route is redistributed. See “Using Route Maps” on page 2-22 for more information.

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

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

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

```
-> show ipv6 redist

Source       Destination
Protocol     Protocol     Status    Route Map
-------------+------------+---------+--------------------
localIPv6    RIPng        Enabled   ipv6rm
OSPFv3       RIPng        Enabled   ospf-to-rip
```

Configuring the Administrative Status of the Route Map Redistribution

The administrative status of a route map redistribution configuration is enabled by default. To change the administrative status, use the **admin-state** parameter with the **ipv6 redist** command. For example, the following command disables the redistribution administrative status for the specified route map:

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

The following command example enables the administrative status:

```
-> ipv6 redist ospf into rip route-map ospf-to-rip admin-state enable
```

Route Map Redistribution Example

The following example configures the redistribution of OSPFv3 routes into a RIPng network using a route map (ospf-to-rip) to filter specific routes:

```
-> ip route-map ospf-to-rip sequence-number 10 action deny
-> ip route-map ospf-to-rip sequence-number 10 match tag 5
-> ip route-map ospf-to-rip sequence-number 10 match route-type external type2
-> ip route-map ospf-to-rip sequence-number 20 action permit
-> ip route-map ospf-to-rip sequence-number 20 match ipv6-interface intf_ospf
-> ip route-map ospf-to-rip sequence-number 20 set metric 255

->ip route-map ospf-to-rip sequence-number 30 action permit
->ip route-map ospf-to-rip sequence-number 30 set tag 8

-> ipv6 redist ospf into rip route-map ospf-to-rip
```

The resulting ospf-to-rip route map redistribution configuration does the following

- Denies the redistribution of Type 2 external OSPF routes with a tag set to five.
- Redistributes into RIPng all routes learned on the intf_ospf interface and sets the metric for such routes to 255.
- Redistributes into RIPng all other routes (those not processed by sequence 10 or 20) and sets the tag for such routes to eight.

Configuring Router Capabilities

The following list shows various commands that can be useful in tailoring a router's performance capabilities. All of the listed parameters have defaults that are acceptable for running an OSPFv3 network.

ipv6 ospf host	Creates and deletes an OSPFv3 entry for directly attached hosts.
ipv6 ospf mtu-checking	Enables or disables the use of Maximum Transfer Unit (MTU) checking on received OSPFv3 database description packets.
ipv6 ospf route-tag	Configures a tag value for Autonomous System External (ASE) routes created.
ipv6 ospf spf-timer	Configures timers for Shortest Path First (SPF) calculation.

To enable MTU checking, enter:

```
-> ipv6 ospf mtu-checking
```

To set the route tag to 5, enter:

```
-> ipv6 ospf route-tag 5
```

To set the SPF timer delay to 3 and the hold time to 6, enter:

```
-> ipv6 ospf spf-timer delay 3 hold 6
```

To return a parameter to its default setting, enter the command with no parameter value, as shown:

```
-> ipv6 ospf spf-timer
```

Configuring Static Neighbors

For non-broadcast networks (NBMA, point-to-point, and point-to-multipoint), the OSPFv3 neighbor has to be manually configured.

NBMA requires all routers attached to the network to communicate directly (unicast), and every attached router in this network becomes aware of all of its neighbors through configuration. It also requires a Designated Router (DR) “eligibility” flag to be set for every neighbor.

To set up a router to use NBMA routing, follow the following steps:

1 Create an OSPFv3 interface using the CLI command **ipv6 ospf interface** and perform all the normal configuration for the interface as with broadcast networks (attaching it to an area, enabling the status, and so on).

2 The OSPFv3 interface type for this interface must be set to non-broadcast using the CLI **ipv6 ospf interface type** command. For example, to set interface vlan-213 to be an NBMA interface, enter the following:

```
-> ipv6 ospf interface vlan-213 type nbma
```

3 Configure static neighbors for every OSPFv3 router in the network using the **ipv6 ospf neighbor** command. For example, to create an OSPFv3 neighbor with a link-local address to be a static neighbor, enter the following:

```
-> ipv6 ospf neighbor fe80::2e0:b1ff:fe7e:5f1e interface vlan-213 eligible
```

The neighbor attaches itself to the right interface by matching the network address of the neighbor and the interface. If the interface has not yet been created, the neighbor gets attached to the interface as and when the interface comes up.

If this neighbor is not required to participate in DR election, configure it as ineligible. The eligibility can be changed at any time as long as the interface it is attached to is in the disabled state.

Configuring Redundant CMMs for Graceful Restart

To enable OSPFv3 graceful restart on OmniSwitch chassis-based switches, use the **ipv6 ospf restart** command. By default, OSPFv3 graceful restart is enabled.

The following command enables both planned and unplanned restarts.

```
-> ipv6 ospf restart
```

To disable OSPFv3 graceful restart, use the **no** form of the **ipv6 ospf restart** command:

```
-> no ipv6 ospf restart
```

Optionally, you can configure graceful restart parameters with the following CLI commands:

ipv6 ospf restart initiate	Initiates a planned OSPFv3 graceful restart.
ipv6 ospf restart-interval	Configures the grace period for achieving a graceful OSPFv3 restart.
ipv6 ospf restart-helper	Administratively enables the capability of an OSPFv3 router to operate in helper mode in response to a router performing a graceful restart.
ipv6 ospf restart-helper strict-lsa-checking	Enables whether or not a changed Link State Advertisement (LSA) will result in termination of graceful restart by a helping router.

For more information about graceful restart commands, see the “OSPFv3 Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

OSPFv3 Application Example

This section will demonstrate how to set up a simple OSPFv3 network. It uses three routers, each with an area. Each router uses three VLANs. A backbone connects all the routers. This section will demonstrate how to set it up by explaining the necessary commands for each router.

The following diagram is a simple OSPFv3 network. It will be created by the steps listed on the following pages.

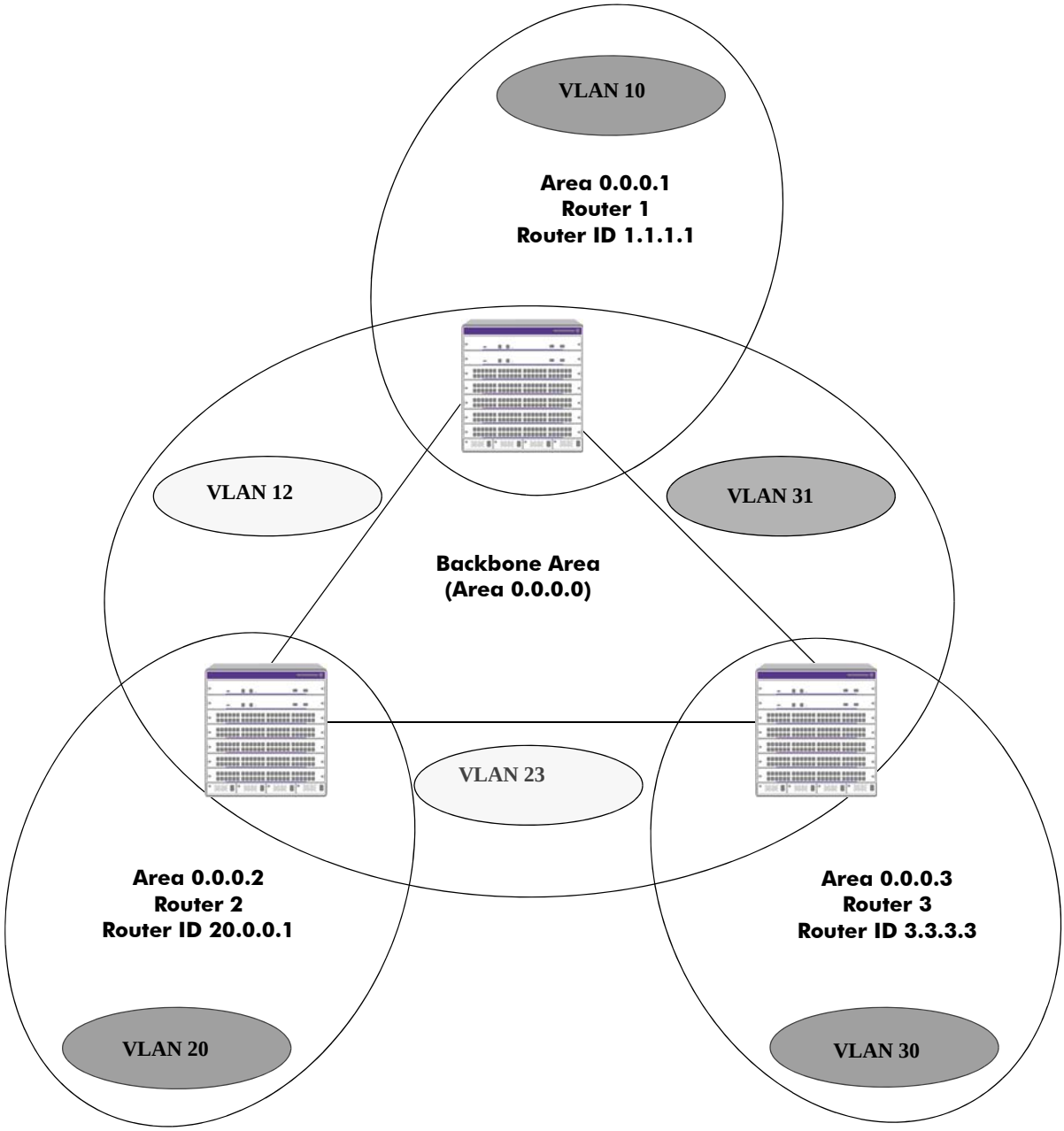


Figure 2-7 : Three Area OSPFv3 Network

Step 1: Prepare the Routers

The first step is to create the VLANs on each router, add an IP interface to the VLAN, assign a port to the VLAN, and assign a router identification number to the routers. For the backbone, the network design in this case uses slot 2, port 1 as the egress port and slot 2, port 2 as ingress port on each router. Router 1 connects to Router 2, Router 2 connects to Router 3, and Router 3 connects to Router 1 using 10/100 Ethernet cables.

Note. The ports will be statically assigned to the router, as a VLAN must have a physical port assigned to it in order for the router port to function. However, the router could be set up in such a way that mobile ports are dynamically assigned to VLANs using VLAN rules. See the chapter titled "Defining VLAN Rules" in the see the *OmniSwitch AOS Release 8 Network Configuration Guide.*

The commands setting up VLANs are shown below:

Router 1 (using ports 2/1 and 2/2 for the backbone, and ports 2/3-5 for end devices):

```
-> vlan 31
-> ipv6 interface vlan-31 vlan 31
-> ipv6 address 2001:1::1/64 vlan-31
-> vlan 31 members port 2/1 untagged

-> vlan 12
-> ipv6 interface vlan-12 vlan 12
-> ipv6 address 2001:2::1/64 vlan-12
-> vlan 12 members port 2/2 untagged

-> vlan 10
-> ipv6 interface vlan-10 vlan 10
-> ipv6 address 2001:3::1/64 vlan-10
-> vlan 10 members port 2/3-5 untagged

-> ip router router-id 1.1.1.1
```

These commands created VLANs 31, 12, and 10.

- VLAN 31 handles the backbone connection from Router 1 to Router 3, using the IP router port `2001:1::1/64` and physical port 2/1.
- VLAN 12 handles the backbone connection from Router 1 to Router 2, using the IP router port `2001:2::1/64` and physical port 2/2.
- VLAN 10 handles the device connections to Router 1, using the IP router port `2001:3::1/64` and physical ports 2/3-5. More ports could be added at a later time if necessary.

The router was assigned the Router ID of 1.1.1.1.

Router 2 (using ports 2/1 and 2/2 for the backbone, and ports 2/3-5 for end devices):

```
-> vlan 12
-> ipv6 interface vlan-12 vlan 12
-> ipv6 address 2001:2::2/64 vlan-12
-> vlan 12 members port 2/1 untagged

-> vlan 23
-> ipv6 interface vlan-23 vlan 23
-> ipv6 address 2001:5::1/64 vlan-23
-> vlan 23 members port 2/2 untagged
```

```
-> vlan 20
-> ipv6 interface vlan-20 vlan 20
-> ipv6 address 2001:4::1/64 vlan-20
-> vlan 20 members port 2/3-5 untagged

-> ipv6 router router-id 2.2.2.2
```

These commands created VLANs 12, 23, and 20.

- VLAN 12 handles the backbone connection from Router 1 to Router 2, using the IP router port `2001:2::2/64` and physical port 2/1.
- VLAN 23 handles the backbone connection from Router 2 to Router 3, using the IP router port `2001:5::1/64` and physical port 2/2.
- VLAN 20 handles the device connections to Router 2, using the IP router port `2001:4::1/64` and physical ports 2/3-5. More ports could be added at a later time if necessary.

The router was assigned the Router ID of 2.2.2.2.

Router 3 (using ports 2/1 and 2/2 for the backbone, and ports 2/3-5 for end devices):

```
-> vlan 23
-> ipv6 interface vlan-23 vlan 23
-> ipv6 address 2001:5::2/64 vlan-23
-> vlan 23 members port 2/1 untagged

-> vlan 31
-> ipv6 interface vlan-31 vlan 31
-> ipv6 address 2001:1::2/64 vlan-31
-> vlan 31 members port 2/2 untagged

-> vlan 30
-> ipv6 interface vlan-30 vlan 30
-> ipv6 address 2001:6::2/64 vlan-30
-> vlan 30 members port 2/3-5 untagged

-> ipv6 router router-id 3.3.3.3
```

These commands created VLANs 23, 31, and 30.

- VLAN 23 handles the backbone connection from Router 2 to Router 3, using the IP router port `2001:5::2/64` and physical port 2/1.
- VLAN 31 handles the backbone connection from Router 3 to Router 1, using the IP router port `2001:1::2/64` and physical port 2/2.
- VLAN 30 handles the device connections to Router 3, using the IP router port `2001:6::2/64` and physical ports 2/3-5. More ports could be added at a later time if necessary.

The router was assigned the Router ID of 3.3.3.3.

Step 2: Load OSPFv3

The next step is to load OSPFv3 on each router. The commands for this step are below (the commands are the same on each router):

```
-> ipv6 load ospf
```

Step 3: Create the Areas and Backbone

Now the areas should be created. In this case, we will create an area for each router, and a backbone (area 0.0.0.0) that connects the areas.

The commands for this step are below:

Router 1

```
-> ipv6 ospf area 0.0.0.0
-> ipv6 ospf area 0.0.0.1
```

These commands created and enabled area 0.0.0.0 (the backbone) and area 0.0.0.1 (the area for Router 1).

Router 2

```
-> ipv6 ospf area 0.0.0.0
-> ipv6 ospf area 0.0.0.2
```

These commands created and enabled Area 0.0.0.0 (the backbone) and Area 0.0.0.2 (the area for Router 2).

Router 3

```
-> ipv6 ospf area 0.0.0.0
-> ipv6 ospf area 0.0.0.3
```

These commands created and enabled Area 0.0.0.0 (the backbone) and Area 0.0.0.3 (the area for Router 3).

Step 4: Create, Enable, and Assign Interfaces

Next, OSPFv3 interfaces must be created, enabled, and assigned to the areas. The OSPFv3 interfaces should have the same interface name as the IPv6 router interfaces created above in “Step 1: Prepare the Routers” on page 2-31.

Router 1

```
-> ipv6 ospf interface vlan-31 area 0.0.0.0
-> ipv6 ospf interface vlan-12 area 0.0.0.0
-> ipv6 ospf interface vlan-10 area 0.0.0.1
```

IPv6 router interface vlan-31 was associated with OSPFv3 interface vlan-31, enabled, and assigned to the backbone. IPv6 router interface vlan-12 was associated with OSPFv3 interface vlan-12, enabled, and assigned to the backbone. IPv6 router interface vlan-10, which connects to end stations and attached network devices, was associated with OSPFv3 interface vlan-10, enabled, and assigned to Area 0.0.0.1.

Router 2

```
-> ipv6 ospf interface vlan-12 area 0.0.0.0
-> ipv6 ospf interface vlan-23 area 0.0.0.0
-> ipv6 ospf interface vlan-20 area 0.0.0.2
```

IPv6 router interface vlan-12 was associated with OSPFv3 interface vlan-12, enabled, and assigned to the backbone. IPv6 router interface vlan-23 was associated with OSPFv3 interface vlan-23, enabled, and assigned to the backbone. IPv6 router interface vlan-20, which connects to end stations and attached network devices, was associated with OSPFv3 interface vlan-20, enabled, and assigned to Area 0.0.0.2.

Router 3

```
-> ipv6 ospf interface vlan-23 area 0.0.0.0
-> ipv6 ospf interface vlan-31 area 0.0.0.0
-> ipv6 ospf interface vlan-30 area 0.0.0.3
```

IPv6 router interface vlan-23 was associated with OSPFv3 interface vlan-23, enabled, and assigned to the backbone. IPv6 router interface vlan-31 was associated with OSPFv3 interface vlan-31, enabled, and assigned to the backbone. IPv6 router interface vlan-30, which connects to end stations and attached network devices, was associated with OSPFv3 interface vlan-30, enabled, and assigned to Area 0.0.0.3.

Step 5: Examine the Network

After the network has been created, you can check the various aspects using show commands:

- For OSPFv3 in general, use the **show ipv6 ospf** command.
- For areas, use the **show ipv6 ospf area** command.
- For interfaces, use the **show ipv6 ospf interface** command.
- To check for adjacencies formed with neighbors, use the **show ipv6 ospf neighbor** command.
- For routes, use the **show ipv6 ospf routes** command.

Verifying OSPFv3 Configuration

To display information about areas, interfaces, virtual links, redistribution, or OPSFv3 in general, use the **show** commands listed in the following table:

show ipv6 ospf	Displays the OSPFv3 status and general configuration parameters.
show ipv6 redist	Displays the route map redistribution configuration.
show ipv6 ospf border-routers	Displays information regarding all or specified border routers.
show ipv6 ospf host	Displays information on directly attached hosts.
show ipv6 ospf lsdb	Displays LSAs in the LSDB associated with each area.
show ipv6 ospf neighbor	Displays information on OSPFv3 non-virtual neighbors.
show ipv6 ospf routes	Displays the OSPFv3 routes known to the router.
show ipv6 ospf virtual-link	Displays virtual link information.
show ipv6 ospf area	Displays either all OSPFv3 areas, or a specified OSPFv3 area.
show ipv6 ospf interface	Displays OSPFv3 interface information.
ipv6 ospf restart	Displays the OSPFv3 graceful restart related configuration and status.

For more information about the resulting displays from these commands, see the “OSPFv3 Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Examples of the **show ipv6 ospf**, **show ipv6 ospf area**, and **show ipv6 ospf interface** command outputs are given in the section “OSPFv3 Quick Steps” on page 2-3.

3 Configuring IS-IS

Intermediate System-to-Intermediate System (IS-IS) is an International Organization for Standardization (ISO) dynamic routing specification.

IS-IS is a shortest path first (SPF), or link state protocol. It is an interior gateway protocol (IGP) that distributes routing information between routers in a single Autonomous System (AS) in IP as well as in OSI environments. IS-IS chooses the least-cost path as the best path. IS-IS is suitable for complex networks with large number of routers since it provides faster convergence where multiple flows to a single destination can be forwarded through one or more interfaces simultaneously.

IS-IS is also an ISO Connectionless Network Protocol (CLNP). It communicates with its peers using the Connectionless Mode Network Service (CLNS) PDU packets, which means that even in an IP-only environment the IS-IS router must have an ISO address. ISO network-layer addressing is done through Network Service Access Point (NSAP) addresses that identify any system in the OSI network.

In This Chapter

This chapter describes the basic components of IS-IS and how to configure them through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, refer the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The configuration procedures described in this chapter include:

- Loading and enabling IS-IS (see page 3-13).
- Creating IS-IS areas (see page 3-14).
- Activate IPv4 or IPv6 Routing and Creating IS-IS Circuit (see page 3-14).
- Enabling IS-IS authentication (see page 3-17).
- Creating redistribution policies using route maps (see page 3-22).
- Enabling M-ISIS Capability (see page 3-32)

For information on creating and managing VLANs, see “Configuring VLANs” in the see the *OmniSwitch AOS Release 8 Network Configuration Guide.*

IS-IS Defaults Table

The following table shows the default settings of the configurable IS-IS parameters.

Parameter Description	Command	Default Value/Comments
Administrative status of IS-IS	**ip isis admin-state**	disabled
Global level of IS-IS	**ip isis level-capability**	Level-1/2
IS-IS authentication type	**ip isis auth-type**	none
Global CSNP authentication	**ip isis csnp-auth**	enabled
Global Hello authentication	**ip isis hello-auth**	enabled
Global PSNP authentication	**ip isis psnp-auth**	enabled
Link State Packet (LSP) timer	**ip isis lsp-lifetime**	1200 seconds
LSP wait interval	**ip isis lsp-wait**	5 seconds (max-wait) 0 (initial-wait) 1 (second-wait)
SPF time interval	**ip isis spf-wait**	10 seconds (max-wait) 1000 milliseconds (initial-wait) 1000 milliseconds (second-wait)
IS-IS Overload state	**ip isis overload**	disabled (Overload state) infinity (timeout interval)
IS-IS Overload state after bootup	**ip isis overload-on-boot**	disabled (Overload state after bootup) infinity (timeout interval)
IS-IS graceful restart	**ip isis graceful-restart**	disabled
IS-IS graceful restart helper mode	**ip isis graceful-restart helper**	enabled
IS-IS system wait-time	**ip isis strict-adjacency-check**	60 seconds
IS-IS adjacency check configuration	**ip isis strict-adjacency-check**	disable
IS-IS authentication check	**ip isis auth-check**	enabled
Authentication type (per IS-IS level)	**ip isis level auth-type**	none
Hello authentication (per IS-IS level)	**ip isis level hello-auth**	enabled
CSNP authentication (per IS-IS level)	**ip isis level csnp-auth**	enabled
PSNP authentication (per IS-IS level)	**ip isis level psnp-auth**	enabled
Wide metrics (per IS-IS level)	**ip isis level wide-metrics-only**	disabled
IPv6 or IPv4 routing in IS-IS	**ip isis activate-ipv6|ipv4**	Both IPv4 and IPv6 routing is enabled
IPv4 or IPv6 IS-IS circuit	**ip isis vlan**	disabled
IS-IS VLAN status	**ip isis vlan admin-state**	disable

Parameter Description	Command	Default Value/Comments
IS-IS VLAN interface type	**ip isis vlan interface-type**	broadcast
Hello authentication (per VLAN)	**ip isis vlan hello-auth-type**	none
CSNP time interval (per VLAN)	**ip isis vlan csnp-interval**	10 seconds (broadcast) 5 seconds (point-to-point)
IS-IS level (per VLAN)	**ip isis vlan level-capability**	Level-1/2
LSP time interval (per VLAN)	**ip isis vlan lsp-pacing-interval**	100 milliseconds
IS-IS passive interface	**ip isis vlan passive**	disabled
Retransmission time of LSP on a point-to-point interface	**ip isis interface retransmit-interval**	5 seconds
Hello authentication for the specified IS-IS level of an IS-IS circuit	**ip isis vlan level hello-auth-type**	none
Hello time interval for the specified IS-IS level an IS-IS circuit	**ip isis vlan level hello-interval**	designated routers: 3 seconds non-designated routers: 9 seconds
Number of missing Hello PDUs from a neighbor	**ip isis vlan level hello-multiplier**	3
Metric value of the specified IS-IS level of an IS-IS circuit	**ip isis vlan level metric**	10
IS-IS passive interface (per IS-IS level)	**ip isis vlan level passive**	disabled
Interface level priority	**ip isis vlan level priority**	64
Multi-topology capability	**ip isis multi-topology**	disabled

IS-IS Quick Steps

The following steps are designed to show the user the necessary set of commands for setting up a router to use IS-IS:

1 Create a VLAN using the **vlan** command. For example:

```
-> vlan 5 name "vlan-5"
```

2 Assign a port to the VLAN using the **vlan** command. For example:

```
-> vlan 5 port default 2/1
```

3 Assign an IP address to the VLAN using the **ip interface** command. For example:

```
-> ip interface vlan-5 address 120.1.4.1 mask 255.0.0.0 vlan 5
```

4 Load IS-IS using the **ip load isis** command. For example:

```
-> ip load isis
```

5 Create an area ID using the **ip isis area-id** command. For example:

```
-> ip isis area-id 49.0001
```

6 Enable IS-IS using the **ip isis admin-state** command. For example:

```
-> ip isis admin-state enable
```

7 Configure IPv6 or IPv4 routing in IS-IS using the **ip isis activate-ipv6|ipv4** command. For example:

```
-> ip isis activate-ipv4
```

8 Configure IPv4 or IPv6 IS-IS circuit on a particular VLAN using the **ip isis vlan** command. For example:

```
-> ip isis vlan 5
-> ip isis vlan 5 address-family v4
```

9 Enables or disables IS-IS on an circuit using the **ip isis vlan admin-state** command. For example:.

```
-> ip isis vlan 5 admin-state enable
```

10 View the IS-IS settings by using the **show ip isis status** command. The output generated is similar to the following:

```
-> show ip isis status
================================================================================
ISIS Status
================================================================================
System Id              : 1000.0000.0107
Admin State            : UP
Protocols Enabled      : IPv4 IPv6
Last Enabled           : Wed May 13 23:36:15 2015
Level Capability       : L1L2
Authentication Check   : True
Authentication Type    : Md5
Graceful Restart       : Enabled
GR helper-mode         : Enabled
LSP Lifetime           : 1200
```

```
LSP Wait              : Max: 5 sec  Initial: 0 sec  Second: 1 sec
Adjacency Check       : Loose
L1 Auth Type          : None
L2 Auth Type          : None
L1 Wide Metrics-only  : Disabled
L2 Wide Metrics-only  : Disabled
L1 LSDB Overload      : Disabled
L2 LSDB Overload      : Disabled
L1 LSPs               : 23
L2 LSPs               : 1
Last SPF              : Thu May 14 18:10:50 2015
SPF Wait              : Max: 10000 ms  Initial: 1000 ms  Second: 1000 ms
Hello-Auth Check      : Enabled
Csnp-Auth Check       : Enabled
Psnp-Auth Check       : Enabled
L1 Hello-Auth Check   : Enabled
L1 Csnp-Auth Check    : Enabled
L1 Psnp-Auth Check    : Enabled
L2 Hello-Auth Check   : Enabled
L2 Csnp-Auth Check    : Enabled
L2 Psnp-Auth Check    : Enabled
Multi-Topology        : Disabled
Auto-Configuration    : Disabled
Area Address          : 00.00

================================================================================
```

11 View the IS-IS IPv4 and IPv6 interface information on a VLAN in the IS-IS database using the **show ip isis vlan** command. The output generated is similar to the following:

```
-> show ip isis vlan
================================================================================
ISIS Vlan
================================================================================
Interface     Address-family  Level  VlanID  Oper-state  Admin-state  L1/L2-Metric
--------------------------------------------------------------------------------
p2p_net1024   ipv4            L1     1024    UP          UP           10/-
v6-None       None            -      1024    -           UP           -
p2p_net1026   ipv4            L1     1026    UP          UP           10/-
v6-None       None            -      1026    -           UP           -
--------------------------------------------------------------------------------
Vlans : 2
================================================================================

-> show ip isis vlan detail
================================================================================
ISIS Vlan
================================================================================
--------------------------------------------------------------------------------
VlanId        : 1024                      Level Capability : L1
Oper State    : UP                        Admin State      : UP
Auth-Type     : None                      Address Families : IPv4
Circuit Id    : 1                         Retransmit Int   : 5
Type          : Broadcast                 LSP Pacing Int   : 100
Mesh Group    : Inactive                  CSNP Int         : 10

  Level       : L1                        Adjacencies      : 0
  Desg IS     : 1000.0000.0107
  Auth Type   : None                      Metric           : 10
```

```
  Hello Timer : 9                          Hello Mult       : 3
  Priority    : 64                         Passive          : No

--------------------------------------------------------------------------------
VlanId        : 1026                       Level Capability : L1
Oper State    : UP                         Admin State      : UP
Auth-Type     : None                       Address Families : IPv4
Circuit Id    : 2                          Retransmit Int   : 5
Type          : Broadcast                  LSP Pacing Int   : 100
Mesh Group    : Inactive                   CSNP Int         : 10

  Level       : L1                         Adjacencies      : 1
  Desg IS     : 1000.0000.0107
  Auth Type   : None                       Metric           : 10
  Hello Timer : 9                          Hello Mult       : 3
  Priority    : 64                         Passive          : No

--------------------------------------------------------------------------------
Vlans : 2
================================================================================
```

IS-IS Overview

IS-IS is an SPF or link state protocol. IS-IS is also an IGP that distributes routing information between routers in a single AS. It supports pure IP and OSI environments, as well as dual environments (both IP and OSI). However, it is deployed extensively in IP-only environments.

IS-IS uses a two-level hierarchy to support large routing domains. A large routing domain may be administratively divided into areas, with each router residing in exactly one area. Routing within an area is referred to as Level-1 routing. A Level-1 Intermediate System (IS) keeps track of routing within its own area. Routing between areas is referred to as Level-2 routing. A Level-2 IS keeps track of paths to destination areas.

IS-IS identifies a device in the network by the NSAP address. NSAP address is a logical point between network and transport layers. It consists of the following three fields:

- **NSEL field—**The N-Selector (NSEL) field is the last byte and it must be specified as a single byte with two hex digits preceded by a period (.). Normally, the NSEL value is set to 00.The NSAP address with its NSEL set to 00 is called Network Entity Title (NET). A NET implies the network layer address of IS-IS.
- **System ID—** This ID occupies the 6 bytes preceding the NSEL field. It is customary to use either a MAC address from the router (for Integrated IS-IS) or an IP address (for example, the IP address of a Loopback interface) as part of the system ID.
- **Area ID—**The area ID occupies the rest of NSAP address.

When a router starts, it uses the IS-IS Hello protocol to discover neighbors and establish adjacencies. The router sends Hello packets through all IS-IS-enabled interfaces to its neighbors, and in turn receives Hello packets. In a broadcast network, the Hello protocol elects a Designated Intermediate System (DIS) for the network.

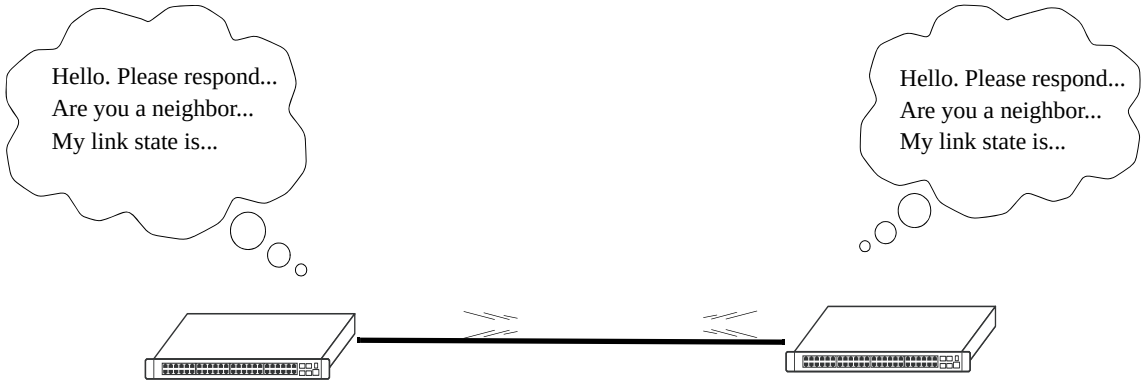


Figure 3-1 : IS-IS Hello Protocol

Separate DISs are elected for Level-1 and Level-2 routing. Election of the DIS is based on the highest interface priority, the default value of which is 64. Priority can also be manually configured, the range being 1–127. In case of a tie, the router with the highest Subnetwork Point Of Attachment (SNPA) address (usually the MAC address) for that interface is elected as the DIS.

Routers that share common data links will become IS-IS neighbors if their Hello packets contain data that meet the requirements for forming an adjacency. The requirements may differ slightly depending on the type of media being used, which is either point-to-point or broadcast. The primary criteria for forming adjacencies are authentication match, IS-type, and MTU size.

Adjacencies control the distribution of routing protocol packets. Routing protocol packets are sent and received only on adjacencies. In particular, distribution of topological database updates proceeds along adjacencies.

After establishing adjacencies, routers will build a link-state packet (LSP) based upon their local interfaces that are configured for IS-IS and prefixes learned from other adjacent routers. Routers flood LSPs to all adjacent neighbors except the neighbor from which they received the same LSP. Routers construct their link-state database from these packets.

The link state is also advertised when a router's state changes. A router'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.

Link state packets are flooded throughout the AS. The flooding algorithm ensures that all routers have exactly the same topological database. This database consists of a collection of link state packets received from each router belonging to the area. From this database, each router calculates the shortest-path tree, with itself as the root. This shortest-path tree, in turn, yields a routing table for the protocol.

IS-IS Packet Types

IS-IS transmits data in little chunks known as packets. There are four packet types in IS-IS. They are:

- **Intermediate System-to-Intermediate System Hello (IIH)—**Used by routers to detect neighbors and form adjacencies.
- **Link State Packet (LSP)**—Contains all the information about adjacencies, connected IP prefixes, OSI end system, area address, etc. There are four types of LSPs: Level-1 pseudo node, Level-1 non-pseudo node, Level-2 pseudo node, and Level-2 non-pseudo node.
- **Complete Sequence Number PDU (CSNP)**—Contains a list of all the LSPs from the current database. CSNPs are used to inform other routers about LSPs that may be outdated or missing from their own database. This ensures that all routers have the same information and are synchronized.
- **Partial Sequence Number PDU (PSNP)**—Used to request an LSP(s) and acknowledge receipt of an LSP(s).

IS-IS Areas

IS-IS allows collections of contiguous networks and hosts to be grouped together as an *area*. Each area runs a separate copy of the basic link state routing algorithm (usually called SPF). This means that each area has its own topological database as explained in the previous section.

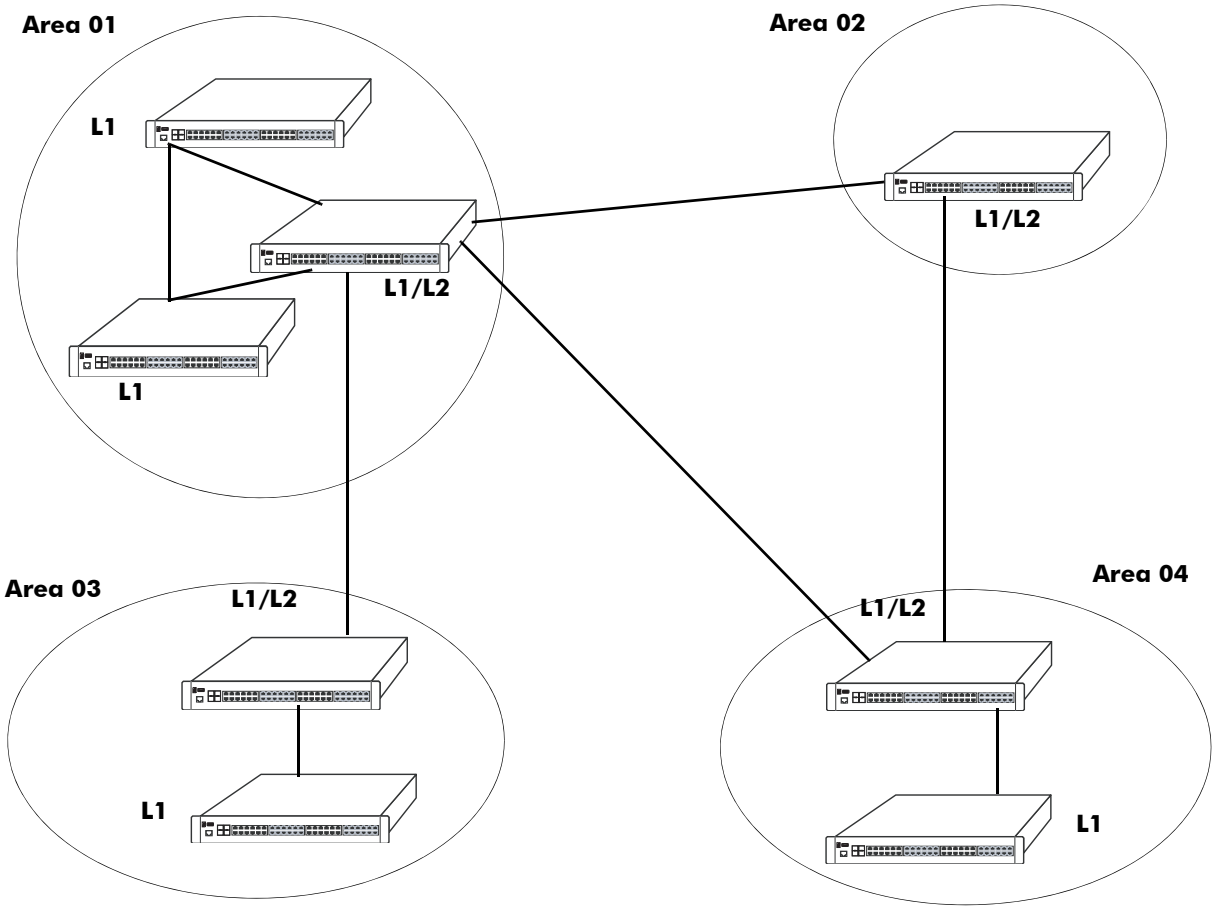


Figure 3-2 : IS-IS Areas

An area's topology is visible only to the members of that area. Routers inside a given area do not know the detailed topology outside the area. This isolation of knowledge enables the protocol to reduce routing traffic by concentrating on small areas of an AS, as compared to treating the entire AS as a single link state domain. In IS-IS, the router belongs entirely to a single area.

When an AS is split into IS-IS areas, the routers are classified into the following three categories:

- **Level-1 routers**—These are Intra-area routers and form relationship with other Level-1 or Level-1/2 routers within the same area.
- **Level-1/2 routers**—These routers form relationship with other Level-1, Level-2, or Level-1/2 routers. They are used to connect Inter-area routers with Intra-area routers.
- **Level-2 routers**—These are Inter-area routers and form relationship with other Level-2 or Level-1/2 routers

These Level capabilities can be defined globally on a router or on specific interfaces. Since a separate copy of the link state algorithm is run in each area, most configuration parameters are defined on a per-router basis. All routers belonging to an area must agree on that area's configuration. Misconfiguration will keep neighbors from forming adjacencies between themselves, and IS-IS will not function.

Graceful Restart on Stacks with Redundant Switches

OmniSwitch stacks with two or more switches support redundancy; if the primary switch fails or goes offline, the secondary switch is instantly notified. The secondary switch automatically assumes the primary role. This transition from secondary to primary is known as *takeover*.

When the router is in the graceful restart mode, it informs its neighbors of the restart. The IS-IS Hello (IIH) messages are modified to signal a graceful restart request. The neighbors respond by sending back their own IIHs with an acknowledgment of the restart, along with a "Remaining Time" value to indicate how long they will wait for a restart. The neighbors also continue to send out LSPs with the restarting router still listed as an adjacency, thus avoiding SPF calculations and enabling traffic to flow to the router from neighbors.

The restarting router continues to forward LSPs using its pre-restart forwarding tables. When graceful restart is enabled, the router can either be a helper or a restarting router, or both. Only helper mode is supported. If a helper is enabled on a neighbor, it begins the Link State Database synchronization process. They send their Complete Sequence Number PDUs (CSNPs) to the restarting router. The restarting router can then determine the LSPs it needs and request them. After it receives all requested LSPs, the database is synchronized.

Note. When graceful restart is enabled on the router, the helper mode is automatically enabled by default.

When the graceful restart timer expires, the restarting router runs the SPF calculation to re-compute IS-IS routes. Only then does it flood LSPs to neighbors and comes back to normal protocol behavior.

In the network illustration below, a helper router, Router Y, monitors the network for topology changes. As long as there are none, it continues to advertise its LSPs as if the restarting router, Router X, had remained in continuous IS-IS operation (i.e., Router Y's LSPs continue to list an adjacency to Router X over network segment S, regardless of the adjacency's current synchronization state).

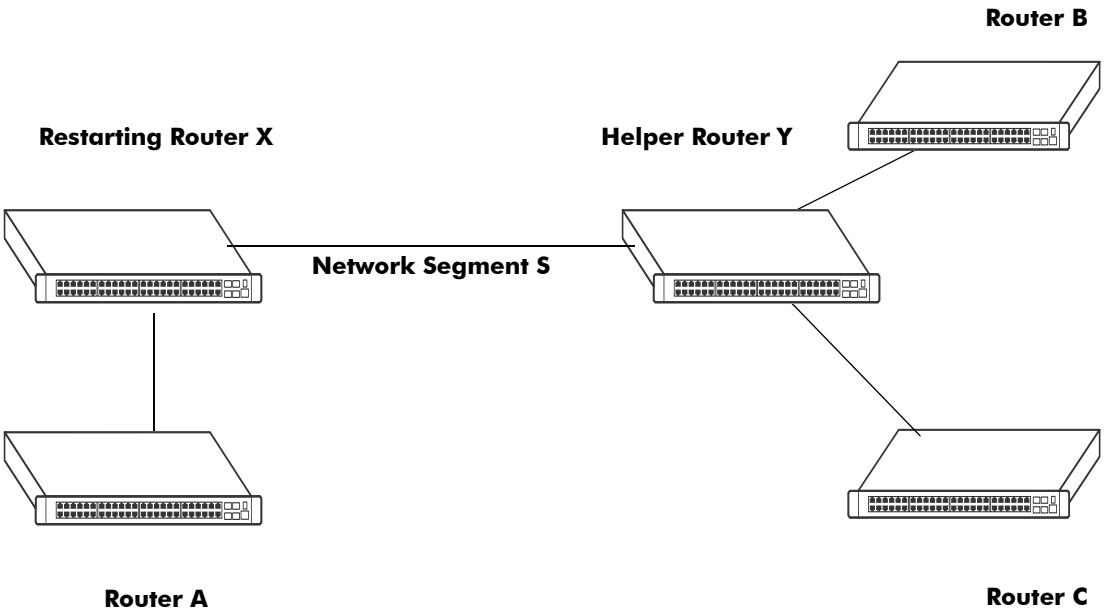


Figure 3-3 : IS-IS Graceful Restart Helper and Restarting Router

If the restarting router, Router X, is identified as the Designated Router (DIS) on the network segment S at the beginning of the helping relationship, the helper neighbor, Router Y, will maintain Router X as the DIS until the helping relationship is terminated. If there are multiple adjacencies with the restarting Router X, Router Y will act as a helper on all other adjacencies.

Configuring IS-IS

Configuring IS-IS on a router requires several steps. Depending on your requirements, you may need to perform all the steps listed below.

By default, IS-IS is disabled on the router. Configuring IS-IS consists of the following tasks:

- Set up the basics of the IS-IS network by configuring the required VLANs and assigning ports to the VLANs. This is described in "Preparing the Network for IS-IS" on page 3-13.
- Enable IS-IS. When the image file for advanced routing is installed, you must load the code and enable IS-IS. The commands for enabling IS-IS are described in "Activating IS-IS" on page 3-13.
- Configure an IS-IS area ID. The commands to create areas and backbones are described in "Creating an IS-IS Area ID" on page 3-14.
- Activate IPv6 or IPv4 routing in IS-IS. This is described in "Activate IPv4 or IPv6 Routing" on page 3-14.
- Configure an IPv4/IPv6 IS-IS circuit on particular VLAN. This is used to enable IS-IS routing on a particular VLAN. Creating IS-IS circuit is described in "Creating IS-IS Circuit" on page 3-14.
- Configure IS-IS levels. Routers are configured at different IS-IS levels. This is described in "Configuring the IS-IS Level" on page 3-15.
- Enable summarization. Routes can be summarized on routers. This is described in "Enabling Summarization" on page 3-16.
- Configure IS-IS authentication (optional). This is described in "Enabling IS-IS Authentication" on page 3-17.
- Configure interface level parameters (optional). The commands to configure interface level parameters are described in "Modifying IS-IS Circuit Parameters" on page 3-21.
- Create a redistribution policy and enable the same using route maps (optional). To create route maps, see "Configuring Redistribution Using Route Maps" on page 3-22.
- Configure router capabilities (optional). There are several commands that influence router operation. These are covered briefly in the table in "Configuring Router Capabilities" on page 3-28.
- Configure redundant switches for graceful IS-IS restart (optional). Configuring switches with redundant switches for graceful restart is described in "Configuring Redundant Switches in a Stack for Graceful Restart" on page 3-28.

At the end of the chapter is a simple IS-IS network diagram with instructions on how it was created on a router-by-router basis. See "IS-IS Application Example" on page 3-29 for more information.

Preparing the Network for IS-IS

IS-IS operates over normal switch functions, using existing ports, virtual ports, VLANs, etc. However, the following network components should already be configured:

- **Configure VLANs that are to be used in the IS-IS network**. VLANs should be created for all the connected devices that will participate in the IS-IS network. VLAN configuration is described in "Configuring VLANs" in the *OmniSwitch AOS Release 8 Network Configuration Guide*.
- **Assign IP interfaces to the VLANs**. IP interfaces or router ports, must be assigned to the VLAN. Assigning IP interfaces is described in "Configuring IP" in the *OmniSwitch AOS Release 8 Network Configuration Guide*.
- **Assign ports to the VLANs**. The physical ports participating in the IS-IS network must be assigned to the created VLANs. Assigning ports to a VLAN is described in "Assigning Ports to VLANs" in the *OmniSwitch AOS Release 8 Network Configuration Guide*.
- **Set the area ID** (optional). The routers participating in the IS-IS network must be assigned an area identification number. The area ID is a part of the Network Service Access Point (NSAP) address, which identifies a point of connection to the network, such as a router interface. The area identification number assignment is discussed in "Creating an IS-IS Area ID" on page 3-14.

Activating IS-IS

To run IS-IS on the router, the advanced routing image must be installed. For information on how to install image files, refer to the *OmniSwitch AOS Release 8 Switch Management Guide*.

After the image file has been installed onto the router, you need to load the IS-IS software into the memory and enable it, as described below:

Loading the Software

To load the IS-IS software into the router's running configuration, enter the **ip load isis** command at the system prompt:

```
-> ip load isis
```

The IS-IS software is now loaded into the memory, and can be enabled. IS-IS is not loaded on the switch by default.

Enabling IS-IS

Once the IS-IS software has been loaded into the router's running configuration (either through the CLI or on startup), it must be enabled. To enable IS-IS on a router, enter the **ip isis admin-state** command at the CLI prompt, as shown:

```
-> ip isis admin-state enable
```

Once IS-IS is enabled, you can begin to set up IS-IS parameters. To disable IS-IS, enter the following:

```
-> ip isis admin-state disable
```

Removing IS-IS

To remove IS-IS from the router memory, it is necessary to manually edit the **boot.cfg** file. The **boot.cfg** file is an ASCII text-based file that controls many of the switch parameters. Open the file and delete all references to IS-IS.

For the operation to take effect the switch needs to be rebooted.

Creating an IS-IS Area ID

IS-IS allows a set of network devices in an AS to be grouped together in *areas*. Each area is identified by an *area ID*. The area ID is a 1–13 byte variable length integer, which specifies the area address of an IS-IS routing process.

For creating an IS-IS area first assign area ID to each router present in the network by using the **ip isis area-id** command. There can be more than one router in an area.

Note. Each router can have a maximum of 3 area IDs assigned to it**.**

Creating an Area ID

To create an area ID and associate it with a router, enter the **ip isis area-id** command with the area identification number at the CLI prompt, as shown:

```
-> ip isis area-id 49.0001
```

Area ID 49.0001 will now be created on the router with the default parameters.

Deleting an Area ID

To delete an area ID, enter the **ip isis area-id** command, as shown:

```
-> no ip isis area-id 49.0001
```

Activate IPv4 or IPv6 Routing

To activate IPv6 or IPv4 routing in IS-IS, enter the **ip isis activate** command as shown. By default, both IPv4 and IPv6 routing is enabled in IS-IS.

```
-> ip isis activate-ipv6
-> ip isis activate-ipv4
```

The **no** form of this command disables the IPv4/IPv6 routing in IS-IS.

```
-> no ip isis activate-ipv4
```

Creating IS-IS Circuit

Configure an IP/IPv6 IS-IS circuit on particular VLAN. Both IPv4 and IPv6 interfaces can be configured on a particular VLAN to the IS-IS circuit. The type of interfaces (IPv4 or IPv6) is controlled by address-family extension.

To create an IPv4/IPv6 IS-IS circuit, use the **ip isis vlan** command. For example, to create an IPv6 circuit, enter:

```
-> ip isis vlan 10 address-family v6
```

To disable IPv4/IPv6 IS-IS circuit on a particular VLAN, use the **no** form of the **ip isis interface**, as shown:

```
-> no ip isis vlan 10 address-family v6
```

Enabling a IS-IS VLAN Circuit

Once the circuit is created, it must be enabled using the **ip isis vlan admin-state** command, as shown:

```
-> ip isis vlan 10 admin-state enable
```

Configuring the IS-IS Level

The Autonomous System is divided into multiple areas to reduce the control traffic and size of routing table. To communicate within an IS-IS area, Level-1 routers are used. To communicate between areas, Level-2 routers are used. A router can be a configured to be a Level-1 router, a Level-2 router, or both.

The level capability can be configured globally on the router or on specific interfaces. By default, the router can operate at both levels.

To modify the level capability of the router globally, use the **ip isis level-capability** command as explained in the following examples:

To configure a router as a Level-1 router, enter:

```
-> ip isis level-capability level-1
```

To configure the router as a Level-2 router, enter:

```
-> ip isis level-capability level-2
```

To configure the router to have both Level-1 and Level-2 capabilities, enter:

```
-> ip isis level-capability level-1/2
```

To modify the level capability of the router on the specified circuit, use the **ip isis vlan level-capability** command as explained in the following examples:

To configure Level-1 capability on VLAN 10, enter:

```
-> ip isis vlan 10 level-capability level-1
```

To configure Level-2 capability on VLAN 10 capability, enter:

```
-> ip isis vlan 10 level-capability level-2
```

To configure both Level-1 and Level-2 capabilities on VLAN 10, enter:

```
-> ip isis vlan 10 level-capability level-1/2
```

When the level capabilities are configured both globally and on per-interface basis, the combination of the two settings will decide the potential adjacency. The rules for deciding the potential adjacency is explained in the following table:

Global Level	Interface Level	Potential Adjacency
Level-1/2	Level-1	Level-1
Level-1/2	Level-2	Level-2

Global Level	Interface Level	Potential Adjacency
Level-1/2	Level-1/2	Level-1 and/or Level-2
Level-1	Level-1	Level-1
Level-1	Level-1	None
Level-1	Level-1/2	Level-1
Level-2	Level-1	None
Level-2	Level-2	Level-2
Level-2	Level-1/2	Level-2

- When the router is globally configured to act at both levels (Level-1/2) and the interface is configured to act at any level, the potential adjacency will be the level adjacency of the interface.
- When the router is globally configured to act at Level-1, the potential adjacency will also be Level-1. If the interface is configured at Level-2 capability, the router will not form potential adjacency with the neighbor.
- When the router is globally configured to act at Level-2, the potential adjacency will also be at Level-2. If the interface is configured at Level-1 capability, the router will not form potential adjacency with the neighbor.

Enabling Summarization

Route summarization in IS-IS reduces the number of routes that a router must maintain, and represents a series of network numbers in a single summary address.

Summarization can also be enabled or disabled when creating an area. IS-IS routes can be summarized into Level-2 from the Level-1 database. It is not possible to summarize IS-IS internal routes at Level-1, although it is possible to summarize external (redistributed) routes. You can summarize level-1, level-2, level-1/2 IS-IS routes. The metric that is used to advertise the summary address is the smallest metric than any of the more specific IP routes.

For example, to summarize the routes between 100.1.1.0/24 and 100.1.100.0/24 into one, enter the following command:

```
-> ip isis summary-address 100.1.0.0/16 level-2
```

To remove the summary address, enter the following:

```
-> no ip isis summary-address 100.1.0.0/16 level-2
```

Note. IS-IS routes are not summarized by default. If you do not specify the level while configuring the summarization, level-1/2 routes are summarized by default.

IS-IS IPv6 route summarization allows users to create aggregate IPv6 addresses that include multiple groups of IPv6 addresses for a given IS-IS level. IPv6 Routes redistributed from other routing protocols also can be summarized. It is similar to the OSPF area-range command. IS-IS route summarization helps to reduce the size of the LSDB and the routing table, and it also helps to reduce the chance of route flapping. IPv6 route summarization supports:

- Level 1, Level 1-2, and Level 2
- Route summarization for the IPv6 routes redistributed from other protocols
- Metric used to advertise the summary address would be the smallest metric of all the more specific IPv6 routes.

For example, to summarize the routes between 4001:1::/64 to 4001:10::/64 into one, enter the following command:

```
-> ip isis summary-address6 4001::/16 level-1
```

To remove the summary address, enter the following:

```
-> no ip isis summary-address6 4001::/16
```

Displaying Summary Address

You can view the details of the IS-IS summary address using the **show ip isis summary-address** and **show ip isis summary-address6** commands:

```
-> show ip isis summary-address
-> show ip isis summary-address6
```

Enabling IS-IS Authentication

IS-IS allows for the use of authentication on a device. When authentication is enabled, only neighbors using the same type of authentication and the matching keys can communicate.

There are two types of authentication: simple, MD5, and Keychain authentication. Simple authentication requires only a text string as a password, while MD5 is a form of encrypted authentication that requires a key and a password, and a keychain is a form of authentication that allows a regular rotation of keys to be used for limited periods of time.

You can use the **key** parameter to configure the password for Simple or MD5 authentication. Alternatively, you can use the **encrypt-key** parameter to configure the password by supplying the encrypted form of the password as the *encrypt-key*. Configuration snapshot always displays the password in an encrypted form. You should use only the *key* parameter during the CLI configuration. If the encrypt-key parameter is used to configure the password through the CLI, then its value should be the same as the one that appears in the configuration snapshot.

You can use the **keychain** parameter to use the configured keychain for authentication. The two remote machines should have same current active key ID and same authentication type. The show commands will always display the key string in an encrypted format.

Notes.

- By default, the authentication is disabled and no authentication type is configured.
- To enable IS-IS authentication, routers participating in ISIS must be configured with either global level authentication or interface level authentication at both ends of the link.

Simple Authentication

Simple authentication works by including the password in the packet. This helps to protect the routers from a configuration mishap.

To enable simple authentication with plain text key on a router, enter the **ip isis auth-type** command, as shown:

```
-> ip isis auth-type simple key 12345
```

Here, only routers with simple authentication and simple key “12345” will be able to use the configured interface.

You can also use the **encrypt-key** parameter to configure the password by supplying the encrypted form of the password.

```
-> ip isis auth-type simple encrypt-key 31fa061a5de5d1a8
```

If the encrypt-key parameter is used to configure the password through the CLI, then its value should be the same as the one that appears in the configuration snapshot.

Note. Only valid system generated values are accepted as *encrypt-key.*

MD5 Authentication

MD5 authentication can be used to protect the system from malicious actions. MD5 authentication can be used to encrypt information sent over the network. MD5 authentication works by using shared secret key. Key is used to sign the packets with an MD5 checksum, so that the packets cannot be forged or tampered with. Since the key is not included in the packet, snooping the key is not possible.

To enable MD5 authentication with plain text key on a router, enter the **ip isis auth-type** command, as shown:

```
-> ip isis auth-type md5 key 12345
```

Here, only routers with MD5 authentication and password “12345” will be able to use the configured interface.

You can also use the **encrypt-key** parameter to configure the password by supplying the encrypted form of the password.

```
-> ip isis auth-type md5 encrypt-key 31fa061a5de5d1a8
```

If the encrypt-key parameter is used to configure the password through the CLI, then its value should be the same as the one that appears in the configuration snapshot..

Note. Only valid system generated values are accepted as *encrypt-key.*

Global Authentication

The authentication check for all the IS-IS PDUs can be enabled or disabled globally by using the **ip isis auth-check** command.

To enable the authentication check for IS-IS PDUs, enter the following:

```
-> ip isis auth-check enable
```

If enabled, IS-IS PDUs that fail to match either of the authentication type and key requirements are rejected.

To disable the authentication check for IS-IS PDUs, enter the following:

```
-> ip isis auth-check disable
```

If disabled, the authentication PDUs are generated and the IS-IS PDUs are authenticated on receipt. An error message will be generated in case of a mismatch; but PDUs will not be rejected.

Note. By default, authentication check is enabled.

IS-IS authentication can be enabled globally for Hello, CSNP, and PSNP packets.

To enable the authentication of Hello PDUs globally, enter the following:

```
-> ip isis hello-auth
```

To enable the authentication of CSNP PDUs globally, enter the following:

```
-> ip isis csnp-auth
```

To enable the authentication of PSNP PDUs globally, enter the following:

```
-> ip isis psnp-auth
```

Level Authentication

You can enable authentication and configure the authentication types for specific IS-IS levels globally using **ip isis level auth-type** command. For example:

```
-> ip isis level 2 auth-type md5 encrypt-key 7a1e441a014b4030
```

The above example configures the authentication type as MD5 for level 2 IS-IS PDUs and the key.

Note. You can configure the authentication of either simple or MD5 type with the password specified either in plain text or in encrypted form. For the explanations about the authentication types and the key types refer **Simple authentication** and **MD5 authentication**.

IS-IS authentication can be enabled for specific IS-IS PDUs such as Hello, CSNP, and PSNP packets at specific IS-IS levels (Level-1, Level-2, or Level-1/2). Enabling authentication on specific IS-IS levels over-rides the global authentication.

To enable the authentication of Hello PDUs for IS-IS Level-1, enter the following:

```
-> ip isis level 1 hello-auth
```

To enable the authentication of CSNP PDUs for IS-IS Level-2, enter the following:

```
-> ip isis level 2 csnp-auth
```

To enable the authentication of PSNP PDUs for IS-IS Level-2, enter the following:

```
-> ip isis level 2 psnp-auth
```

Note. On a point-to-point link with both levels enabled, if no authentication is configured for Level 1, the hello packets are sent without any password regardless of the Level 2 authentication configurations.

IS-IS Circuit Level Authentication

IS-IS authentication can be enabled for Hello packets at a circuit level using **ip isis vlan hello-auth-type** command.

For example, to enable MD5 authentication of Hello PDUs on the IS-IS circuit, enter the following:

```
-> ip isis vlan 10 hello-auth-type md5 key 12345
```

IS-IS authentication can also be enabled for Hello packets at different levels of an IS-IS circuit using **ip isis vlan level hello-auth-type.**

For example, to enable simple authentication of Hello PDUs at Level-2 of an IS-IS circuit, enter the following:

```
-> ip isis vlan 100 level 2 hello-auth-type simple encrypt-key 7a1e441a014b4030
```

Note. Both the **ip isis vlan hello-auth-type** and **ip isis vlan level hello-auth-type** can be configured for the authentication of either simple, MD5, or keychain type with the password specified either in plain text, encrypted form or key. For the explanations about the authentication types and the key types refer **Simple authentication, MD5 authentication, Keychain Authentication.**

Keychain Authentication

Keychain authentication can be applied at a global level, capability level, circuit level, and capability level per circuit.

To enable keychain authentication on a router, enter **ip isis auth-type** command with the configured keychain to be used as shown.

```
-> ip isis auth-type key-chain 2
```

If a keychain is applied globally, the authentication algorithm of its active key will be used for adjacency formation with all peers. Use **ip isis auth-type none** to remove the keychain from IS-IS adjacency configurations.

To enable keychain authentication for specific IS-IS levels, use **ip isis level auth-type** command. For example, to enable the authentication for IS-IS Level-2, enter the following:

```
-> ip isis level 2 auth-type key-chain 1
```

Use **ip isis level auth-type none** to remove the capability level keychain authentication.

To enable keychain authentication at a circuit level, use **ip isis vlan hello-auth-type** command. For example, to enable keychain authentication on the IS-IS circuit, enter the following.

```
-> ip isis vlan 100 hello-auth-type key-chain 1
```

Use **ip isis vlan hello-auth-type none** to remove the circuit level keychain authentication.

To enable keychain authentication at different levels of an IS-IS circuit, use **ip isis vlan level hello-auth-type** command. For example, to enable keychain authentication at Level-2 of an IS-IS circuit, enter the following:

```
-> ip isis vlan 100 level 2 hello-auth-type key-chain 1
```

Use **ip isis vlan level hello-auth-type none** to remove the capability level keychain authentication per circuit.

For more information on configuring keychain management commands, refer to the “Chassis Management and Monitoring Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Modifying IS-IS Circuit Parameters

To configure the interval between the successive Hello PDUs at the given IS-IS level on a circuit, enter the **ip isis vlan level hello-interval** command, as shown:

```
-> ip isis vlan 10 level 1 hello-interval 50
```

To configure the number of Hello PDUs before the router declares the adjacency as down, use the **ip isis vlan level hello-multiplier** command, as shown:

```
-> ip isis vlan 10 level 1 hello-multiplier 10
```

To configure the metric value of the IS-IS level of the circuit, enter the **ip isis vlan level metric** command, as shown:

```
-> ip isis vlan 10 level 1 metric 25
```

Configuring IS-IS circuit as passive at the specified IS-IS level suppresses IS-IS packets from being sent or received on the interface. For example, to configure the interface from receiving Level-1 IS-IS packets, enter the **ip isis vlan passive** command, as shown:

```
-> ip isis vlan 10 level 1 passive
```

Configuring the priority value helps to determine a DIS in a multi-access network. To configure the priority of the IS-IS circuit for the election of a DIS in a multi-access network, enter the **ip isis vlan level priority** command, as shown:

```
-> ip isis vlan 10 level 1 priority 4
```

There are several other parameters that can be modified on the IS-IS circuit. Most of these deal with timer settings.

The following table shows the various interface parameters that can be set:

ip isis vlan csnp-interval	Configures the time interval in seconds to send Complete Sequence Number PDUs (CSNP) from the specified VLAN circuit.
ip isis vlan lsp-pacing-interval	Configures the interval between IS-IS Link State PDUs (LSP) sent from the specified circuit.
ip isis vlan retransmit-interval	Configures the minimum time between Link State PDU (LSP) transmissions on a point-to-point interface.
ip isis vlan interface-type	Configures the IS-IS interface (circuit) type as broadcast or point-to-point.

These parameters can be added any time. In broadcast networks, the DIS sends CSNP packets to maintain database synchronization. For example, to configure the CSNP PDUs time interval to 50 seconds, enter the following:

```
-> ip isis vlan 101 csnp-interval 50
```

To set the LSP interval to 120 seconds, enter the following:

```
-> ip isis vlan 101 lsp-pacing-interval 120
```

To set the LSP retransmit interval to 100 seconds, enter the following:

```
-> ip isis vlan 101 retransmit-interval 100
```

Note. The retransmit interval should be greater than the expected round-trip delay between two devices. This will avoid any needless retransmission of PDUs.

Configuring Redistribution Using Route Maps

It is possible to configure the IS-IS protocol to advertise routes learned from other routing protocols (AS-external routes) into the IS-IS network. Such a process is referred to as route redistribution and is configured using the **ip redist** command.

IS-IS redistribution uses route maps to control how external routes are learned and distributed. A route map consists of one or more user-defined statements that can determine which routes are allowed or denied access to the IS-IS network. In addition a route map may also contain statements that modify route parameters before they are redistributed.

When a route map is created, it is given a name to identify the group of statements that it represents. This name is required by the **ip redist** command. Therefore, configuring IS-IS route redistribution involves the following steps:

1 Create a route map, as described in "Using Route Maps" on page 3-22.

2 Configure redistribution to apply a route map, as described in "Configuring Route Map Redistribution" on page 3-26.

Using Route Maps

A route map specifies the criteria that are used to control redistribution of routes between protocols. Such criteria are defined by configuring route map statements. There are three different types of statements:

- **Action**—An action statement configures the route map name, sequence number, and whether or not redistribution is permitted or denied based on route map criteria.
- **Match**—A match statement specifies criteria that a route must match. When a match occurs, then the action statement is applied to the route.
- **Set**—A set statement is used to modify route information before the route is redistributed into the receiving protocol. This statement is applied only if all the criteria of the route map is met and the action permits redistribution.

The **ip route-map** command is used to configure route map statements and provides the following **action**, **match**, and **set** parameters:

ip route-map action ...	ip route-map match ...	ip route-map set ...
permit **deny**	**ip address** **ip next-hop** **ipv6 address** **ipv6 next-hop** **tag** **ipv4-interface** **ipv6-interface** **metric** **route-type**	**metric** **metric-type** **tag** **community** **local-preference** **level** **ip-nexthop** **ipv6-nexthop**

Note. The tag parameter is not supported in the current release.

Refer to the “IP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the **ip route-map** command parameters and usage guidelines.

Once a route map is created, it is then applied using the **ip redist** command. See “Configuring Route Map Redistribution” on page 3-26 for more information.

Creating a Route Map

When a route map is created, it is given a name (up to 20 characters), a sequence number, and an action (permit or deny). Specifying a sequence number is optional. If a value is not configured, then the number 50 is used by default.

To create a route map, use the **ip route-map** command with the **action** parameter. For example,

```
-> ip route-map rip-to-isis sequence-number 10 action permit
```

The above command creates the rip-to-isis route map, assigns a **sequence number** of 10 to the route map, and specifies a **permit** action.

To optionally filter routes before redistribution, use the **ip route-map** command with a **match** parameter to configure match criteria for incoming routes. For example,

```
-> ip route-map rip-to-isis sequence-number 10 match metric 8
```

The above command configures a match statement for the rip-to-isis route map to filter routes based on their metric value. When this route map is applied, only RIP routes with a metric value of eight are redistributed into the IS-IS network. All other routes with a different metric value are dropped.

Note. Configuring match statement is not required. However, if a route map does not contain any match statement and the route map is applied using the **ip redist** command, the router redistributes *all* routes into the network of the receiving protocol.

To modify route information before it is redistributed, use the **ip route-map** command with a **set** parameter. For example,

```
-> ip route-map rip-to-isis sequence-number 10 set metric 5
```

The above command configures a set statement for the rip-to-isis route map that changes the metric value to five. Because this statement is part of the rip-to-isis route map, it is only applied to routes that have an existing metric value equal to eight.

The following is a summary of the commands used in the above examples:

```
-> ip route-map rip-to-isis sequence-number 10 action permit
-> ip route-map rip-to-isis sequence-number 10 match metric 8
-> ip route-map rip-to-isis sequence-number 10 set metric 5
```

To verify a route map configuration, use the **show ip route-map** command:

```
-> show ip route-map
Route Maps: configured: 1 max: 200
Route Map: rip-to-isis Sequence Number: 10 Action permit
  match metric 8
  set metric 5
```

Deleting a Route Map

Use the **no** form of the **ip route-map** command to delete an entire route map, a route map sequence, or a specific statement within a sequence.

To delete an entire route map, enter **no ip route-map** followed by the route map name. For example, the following command deletes the entire route map named rip-to-isis:

```
-> no ip route-map rip-to-isis
```

To delete a specific sequence number within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the actual number. For example, the following command deletes sequence 10 from the rip-to-isis route map:

```
-> no ip route-map rip-to-isis sequence-number 10
```

Note that in the above example, the rip-to-isis route map is not deleted. Only those statements associated with sequence 10 are removed from the route map.

To delete a specific statement within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the sequence number for the statement, then either **match** or **set** and the match or set parameter and value. For example, the following command deletes only the match metric 8 statement from route map rip-to-isis sequence 10:

```
-> no ip route-map rip-to-isis sequence-number 10 match metric 8
```

Configuring Route Map Sequences

A route map may consist of one or more sequences of statements. The sequence number determines which statements belong to which sequence and the order in which sequences for the same route map are processed.

To add match and set statements to an existing route map sequence, specify the same route map name and sequence number for each statement. For example, the following series of commands creates route map rm_1 and configures match and set statements for the rm_1 sequence 10:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match metric 8
-> ip route-map rm_1 sequence-number 10 set metric 2
```

To configure a new sequence of statements for an existing route map, specify the same route map name but use a different sequence number. For example, the following command creates a new sequence 20 for the rm_1 route map:

```
-> ip route-map rm_1 sequence-number 20 action permit
-> ip route-map rm_1 sequence-number 20 match ipv4-interface to-finance
-> ip route-map rm_1 sequence-number 20 set metric 5
```

The resulting route map appears as follows:

```
-> show ip route-map rm_1
Route Map: rip-to-isis Sequence Number: 10 Action permit
  match metric 8
  set metric 2
Route Map: rip-to-isis Sequence Number: 20 Action permit
  match ipv4 interface to-finance
  set metric 5
```

Sequence 10 and sequence 20 are both linked to route map rm_1 and are processed in ascending order according to their sequence number value. Note that there is an implied logical OR between sequences. As a result, if there is no match for the metric value in sequence 10, then the match interface statement in sequence 20 is processed. However, if a route matches the metric value 8, then sequence 20 is not used. The set statement for whichever sequence was matched is applied.

A route map sequence may contain multiple match statements. If these statements are of the same kind (e.g., match metric 5, match metric 8, etc.) then a logical OR is implied between each like statement. If the match statements specify different types of matches (e.g. match metric 8, match ip4 interface to-finance, etc.), then a logical AND is implied between each statement. For example, the following route map sequence will redistribute a route if its metric value is either 8 or 5:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match metric 5
-> ip route-map rm_1 sequence-number 10 match metric 8
```

The following route map sequence will redistribute a route if the route has a metric of 8 or 5 *and* if the route was learned on the IPv4 interface to-finance:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match metric 5
-> ip route-map rm_1 sequence-number 10 match metric 8
-> ip route-map rm_1 sequence-number 10 match ipv4-interface to-finance
```

Configuring Access Lists

An IP access list provides a convenient way to add multiple IPv4 addresses to a route map. Using an access list avoids having to enter a separate route map statement for each individual IP address. Instead, a single statement is used that specifies the access list name. The route map is then applied to all the addresses contained within the access list.

Configuring an IP access list involves two steps: creating the access list and adding IP addresses to the list. To create an IP access list, use the **ip access-list** and specify a name to associate with the list. For example,

```
-> ip access-list ipaddr
```

To add addresses to an access list, use the **ip access-list address** command. For example, the following commands add addresses to an existing access list:

```
-> ip access-list ipaddr address 16.24.2.1/16
```

Use the same access list name each time the above commands are used to add additional addresses to the same access list. In addition, both commands provide the ability to configure if an address and/or its matching subnet routes are permitted (the default) or denied redistribution. For example:

```
-> ip access-list ipaddr address 16.24.2.1/16 action deny redist-control all-
subnets
```

For more information about configuring access list commands, see the “IP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Route Map Redistribution

The **ip redist** command is used to configure the redistribution of routes from a source protocol into the IS-IS destination protocol. This command is used on the IS-IS router that will perform the redistribution.

A source protocol is a protocol from which the routes are learned. A destination protocol is the one into which the routes are redistributed. Make sure that both protocols are loaded and enabled before configuring redistribution.

Redistribution applies criteria specified in a route map to routes received from the source protocol. Therefore, configuring redistribution requires an existing route map. For example, the following command configures the redistribution of RIP routes into the IS-IS network using the rip-to-isis route map:

```
-> ip redist rip into isis route-map rip-to-isis
```

RIP routes received by the IS-IS router interface are processed based on the contents of the rip-to-isis route map. Routes that match criteria specified in this route map are either allowed or denied redistribution into the IS-IS network. The route map may also specify the modification of route information before the route is redistributed. See “Using Route Maps” on page 3-22 for more information.

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

```
-> no ip redist rip into isis route-map rip-to-isis
```

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

```
-> show ip redist
Source       Destination
Protocol     Protocol     Status    Route Map
-------------+------------+---------+--------------------
OSPF         ISIS         Enabled   ospf-to-isis
RIP          ISIS         Enabled   rip-to-isis
```

Configuring the Administrative Status of the Route Map Redistribution

The administrative status of a route map redistribution configuration is enabled by default. To change the administrative status, use the **status** parameter with the **ip redist into isis route-map** command. For example, the following command disables the redistribution administrative status for the specified route map:

```
-> ip redist rip into isis route-map rip-to-isis status disable
```

The following command example enables the administrative status:

```
-> ip redist rip into isis route-map rip-to-isis status enable
```

Route Map Redistribution Example

The following example configures the redistribution of RIP routes into an IS-IS network using a route map (rip-to-isis) to filter specific routes:

```
-> ip route-map rip-to-isis sequence-number 10 action deny
-> ip route-map rip-to-isis sequence-number 10 match metric 5
-> ip route-map rip-to-isis sequence-number 20 action permit
-> ip route-map rip-to-isis sequence-number 20 match ipv4-interface intf_isis
-> ip route-map rip-to-isis sequence-number 20 set metric 60

-> ip route-map rip-to-isis sequence-number 30 action permit
-> ip route-map rip-to-isis sequence-number 30 set metric 8
-> ip redist rip into isis route-map rip-to-isis
```

The resulting rip-to-isis route map redistribution configuration does the following:

- Denies the redistribution of RIP routes with a metric value set to five.
- Redistributes into IS-IS all routes learned on the intf_isis interface and sets the metric for such routes to 60.

Note. Wide metrics need to be enabled, if a metric of more than 64 is configured.

- Redistributes all other routes (those not processed by sequence 10 or 20) and sets the metric for such routes to eight.

IS-IS allows redistributing Level-1 IS-IS routes into Level-2 IS-IS routes. This is termed as Level-1 to Level-2 Leaking. This release also supports the prefix distribution from the level-2 IS-IS routes to level-1 IS-IS routes.

The following example configures the IS-IS Level-1 to Level-2 Leaking routes using a route map (is2is) to filter specific routes.

To redistribute IS-IS Level-1 routes into IS-IS Level-2 routes, use the following route map sequence:

```
-> ip route-map is2is sequence-number 1 action permit
-> ip route-map is2is sequence-number 1 match route-type level1
-> ip route-map is2is sequence-number 1 set level level2
-> ip redist isis into isis route-map is2is status enable
```

The resulting is2is route map redistribution configuration redistributes all Level-1 IS-IS routes into Level-2 IS-IS routes.

Configuring Router Capabilities

The following table lists various commands that can be useful in tailoring a router's performance capabilities. All the listed parameters have defaults that are acceptable for running an IS-IS network.

ip isis overload	Sets the IS-IS router to operate in the overload state.
ip isis overload-on-boot	Configures the router to be in the overload state.
ip isis strict-adjacency-check	Enables or disables the adjacency check configuration.

To set the IS-IS router to operate in overload state, enter:

```
-> ip isis overload timeout 70
```

To configure the router to be in the overload state, enter:

```
-> ip isis overload-on-boot timeout 80
```

To enable the adjacency check configuration, enter:

```
-> ip isis strict-adjacency-check enable
```

Configuring Redundant Switches in a Stack for Graceful Restart

By default, IS-IS graceful restart is disabled. When graceful restart is enabled, the router can either be a helper or a restarting router. When graceful restart is enabled on the router, the helper mode is automatically enabled by default. To configure IS-IS graceful restart support on OmniSwitch switches, use the **ip isis graceful-restart** command.

Note. In the current release, only the graceful restart helper mode is supported.

For example, to configure graceful restart on the router, enter:

```
-> ip isis graceful-restart
```

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

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

To disable support for graceful restart, use the **no** form of the **ip isis graceful-restart** command by entering:

```
-> no ip isis graceful-restart
```

Continuous forwarding during a graceful restart depends on several factors. If the secondary module has a different router MAC than the primary module, or if one or more ports of a VLAN belonged to the primary module, spanning tree re-convergence might disrupt forwarding state, even though IS-IS performs a graceful restart.

Note. Graceful restart is only supported on active ports (i.e., interfaces), which are on the secondary or idle switches in a stack during a takeover. It is not supported on ports on a primary switch in a stack.

IS-IS Application Example

This section will demonstrate how to set up a simple IS-IS network. It uses two routers, each with an area. Each router is a L1-L2 capable router and can communicate with different areas. This section will demonstrate how to set it up by explaining the necessary commands for each router.

The following diagram is a simple IS-IS network. This network will be created using the steps explained below.

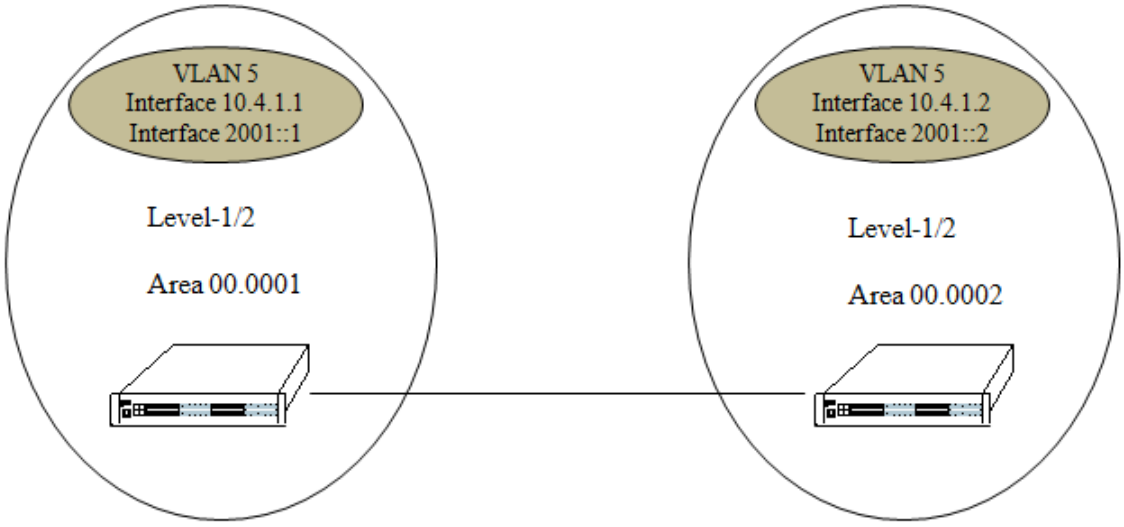


Figure 3-4 : Simple IS-IS Network

Step 1: Prepare the Routers

The first step is to create the VLANs on each router, add IP interface to the VLAN, and assign port to the VLAN.

Note. The ports will be statically assigned to the router, as a VLAN must have a physical port assigned to it for the router port to function. However, the router could be set up in such a way that mobile ports are dynamically assigned to VLANs using VLAN rules. See the chapter titled “Defining VLAN Rules” in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

The commands to setup VLANs are shown below:

Router 1

```
-> vlan 5 name vlan-isis
-> ip interface vlan-isis address 10.4.1.1 mask 255.0.0.0 vlan 5
-> vlan 5 port default 1/10
-> ipv6 interface vlan-isis vlan 5
-> ipv6 address 2001::1/64 vlan-isis
```

Router 2

```
-> vlan 5 name vlan-isis
-> ip interface vlan-isis address 10.4.1.2 mask 255.0.0.0 vlan 5
-> vlan 5 port default 1/10
-> ipv6 interface vlan-isis vlan 5
-> ipv6 address 2001::2/64 vlan-isis
```

Step 2: Enable IS-IS

The next step is to load and enable IS-IS on each router. The commands for this are shown below (the commands are the same on each router):

```
-> ip load isis
-> ip isis admin-state enable
```

Step 3: Create and Enable Area ID

Now the areas should be created and enabled. The commands for this are shown below:

Router 1

```
-> ip isis area-id 00.0001
```

This command created the area for Router 1.

Router 2

```
-> ip isis area-id 00.0002
```

This command created the area for Router 2.

Step 4: Configuring IS-IS Level Capability

The router must be configured with the IS-IS level capability, which decides whether the router will route traffic within an area or between two or more areas.

Router 1

```
-> ip isis level-capability level-1/2
```

Router 2

```
-> ip isis level-capability level-1/2
```

Note. The default IS-IS level capability is Level-1/2.

Step 5: Enabling IS-IS on VLAN

Router 1

```
-> ip isis vlan vlan-isis
-> ip isis vlan vlan-isis address-family v4v6
-> ip isis vlan vlan-isis admin-state enable
```

Router 2

```
-> ip isis vlan vlan-isis
-> ip isis vlan vlan-isis address-family v4v6
-> ip isis vlan vlan-isis status enable
```

Step 6: Examine the Network

After the network has been created, you can check various aspects of it using show commands:

- For IS-IS in general, use the **show ip isis statistics** command.
- For SPF details, use the **show ip isis spf** command.

- For summarization details, use the **show ip isis summary-address** command.
- To check for adjacencies formed with neighbors, use the **show ip isis adjacency** command.
- For routes, use the **show ip isis routes** command.
- For details of the interfaces, use the **show ip isis vlan** command.

Multi-Topology IS-IS Overview

Multi-topology (M-ISIS) support is necessary in IS-IS to support network domains in which non-dual stack IS-IS routers exist. The default protocol behavior of IS-IS is to construct shortest paths through the network using the routers' MAC addresses with no regard to the different IP address families supported. This behavior may result in black-holed routing when there are some IPv4-only or IPv6-only routers in an IS-IS routing domain, instead of all dual-stack routers.

M-ISIS mechanism runs multiple, independent IP topologies within a single IS-IS network domain, using separate topology-specific SPF computation and multiple Routing Information Bases (RIBs).

M-ISIS routers advertise their MT capability by including a set of MT TLVs in their Hello PDUs. When the router originates LSPs, it uses MT Reachable IS TLVs to list the topologies the router belongs to and the neighbors in the same topologies. Any IS-IS router that does not advertise MT capability, or does not use the MT TLVs is considered as belonging to the default topology.

On point-to-point interfaces, if two neighboring MT capable IS-IS routers have no common topologies in common, no adjacency is formed. On broadcast interfaces, an adjacency is formed between two or more neighboring IS-IS routers even if there is no topology in common.

Note. M-ISIS is advised in networks containing ISIS enabled routers with a combination of IPv4 and IPv6 capabilities.

M-ISIS Operation

Each Multi-topology runs its own SPF computation. The results of the SPF computation stored in a separate Routing Information Base (RIB), identified by the MT ID.

By default, as per normal IS-IS protocol behavior, SPF computation for IPv4 and IPv6 prefixes results in a common SPF tree. This approach holds true when all IS-IS routers in the network are dual-stack and share a common topology (that is, IPv4 and IPv6 traffic traversing the same links, using corresponding IP interfaces configured on the VLANs). This computation is the non MT-capable mode of IS-IS operation.

If M-ISIS capability is enabled, different SPF computations are performed for IPv4 and IPv6 based on the different MT IDs. Specifically, one common IPv4 SPF computation is performed for the default behavior as well as IPv4 prefixes learned in the context of MT ID 0. IPv6 SPF computation is performed for IPv6 prefixes learned in the context of MT ID 2. In this MT-specific computation, IPv6 prefixes learned through the default behavior (that is, if advertised in IP6 Reachable Prefixes TLV type 236), is omitted from this computation.

Enabling M-ISIS Capability

Use the following command to enable M-ISIS capability support for IS-IS. If enabled, IPv6 SPF computation is performed separate from the IPv4 SPF computation.

```
-> ip isis multi-topology
```

Changing the multi-topology mode with this command will result in internal disabling and re-enabling of IS-IS protocol, with the new mode of operation. This causes IS-IS adjacencies to be reset.

If M-ISIS mode of operation is not enabled, AOS IS-IS operates in a dual-stack mode, computing a single SPF for IPv4 and IPv6. M-ISIS TLVs received in this mode are not processed for SPF calculations.

For backwards compatibility with non M-ISIS aware routers, even if M-ISIS capability is enabled, AOS IS-IS will continue to exchange IPv4 prefixes in the default IPv4 reachability TLVs (and not in the M-IS-IS TLVs in MT ID 0). SPF processing for IPv4 will include default IPv4 Reachability TLVs along with those received in MT ID 0 TLVs (if any).

Verify M-ISIS Configuration

Use the **show ip isis status** and **show ip isis adjacency** commands to view the M-ISIS configuration details.

M-ISIS Configuration Scenario

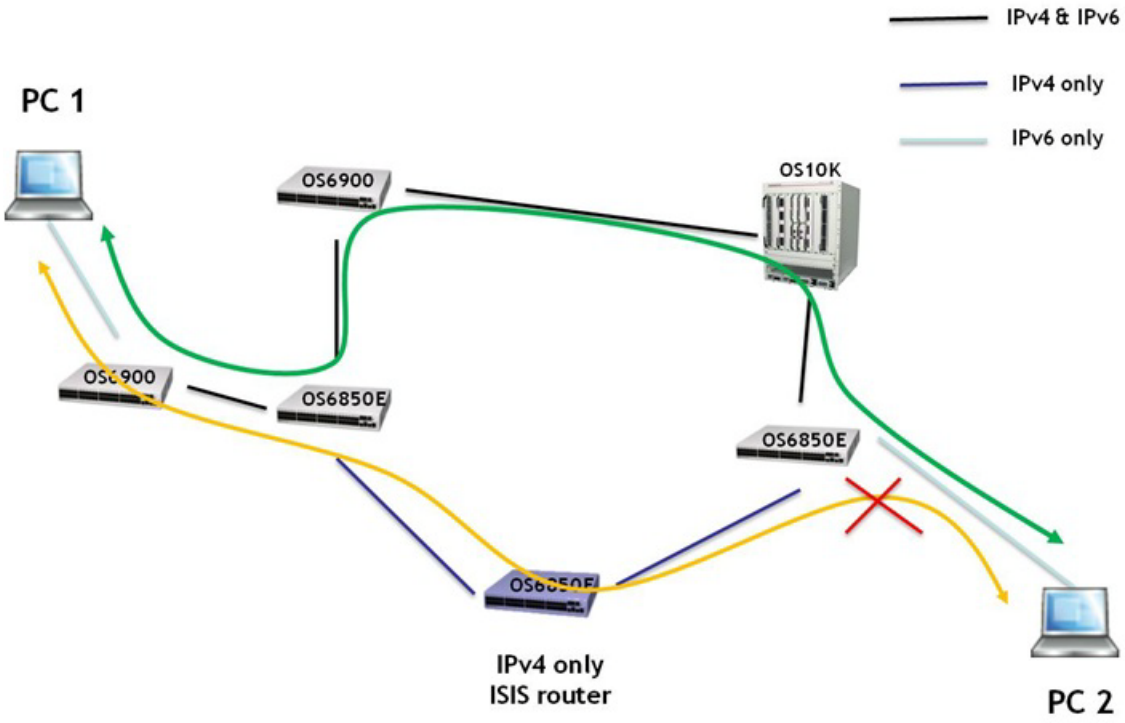


Figure 3-5 : M-ISIS Configuration Scenario

In the above scenario, when M-ISIS is configured, the IPV6 traffic sent from PC2 to PC1 is sent along the green path even though orange path is the shortest path with minimum hops. M-ISIS distinguishes between IPv4 and IPv6 topologies, the dual-stack routers performs SPF calculation separately for IPv4 and IPv6. That is, the MT-specific SPF calculation for IPv6 excludes router IPV4 only router thereby preventing the black-holing.

Verifying IS-IS Configuration

To verify information about adjacent routers, summary-address, SPF, or IS-IS in general, use the **show** commands listed in the following table:

show ip isis adjacency	Displays information about IS-IS adjacent routers.
show ip isis database	Displays IS-IS LSP database information of the adjacent routers.

show ip isis hostname	Displays the database of IS-IS host names.
show ip isis routes	Displays the IS-IS route information known to the router.
show ip isis spf	Displays the IS-IS SPF calculation information.
show ip isis spf-log	Displays the IS-IS SPF log.
show ip isis statistics	Displays the IS-IS statistics information.
show ip isis status	Displays the IS-IS status.
show ip isis summary-address	Displays the IS-IS summary address database.
show ip isis summary-address6	Displays the IS-IS IPv6 summary address database.
show ip redist	Displays the IS-IS configured redistributions.
show ip isis vlan	Displays the IS-IS IPv4 and IPv6 interface information on a VLAN in the IS-IS database.

For more information about the commands output, see Chapter 3, “Configuring IS-IS” in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

4 Configuring BGP

The Border Gateway Protocol (BGP) is an exterior routing protocol that guarantees the loop-free exchange of routing information between autonomous systems. The OmniSwitch implementation supports BGP version 4 and the RFCs specified below.

This chapter describes the configuration and use of BGP in IPv4 and IPv6 environments using the Command Line Interface (CLI). The OmniSwitch implementation of BGP-4 and Multiprotocol Extensions to BGP-4 is based on several RFCs listed below. CLI commands are used in the configuration examples in this chapter. For more details about the syntax of these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Notes.

- In this document, the BGP terms “peer” and “neighbor” are used interchangeably.
- This implementation of BGP allows the configuration and management of BGP in IPv4 and IPv6 environments through CLI, WebView, and SNMP interfaces.

In This Chapter

The topics and configuration procedures in this chapter include:

- Setting up global BGP parameters, such as a router's Autonomous System (AS) number and default local preference. See "Setting Global BGP Parameters" on page 4-18.
- Configuring a BGP peer and setting various parameters on that peer, such as timers, soft reconfiguration, and policies. See "Configuring a BGP Peer" on page 4-24.
- Configuring the advertising of IPv4 routes for IPv4 BGP peer. See "Configuring the advertising of IPv4 routes for an IP BGP peer" on page 4-29.
- Configuring route dampening parameters for the router. See "Controlling Route Flapping Through Route Dampening" on page 4-34.
- Configuring route reflection using single and multiple route reflectors. See "Setting Up Route Reflection" on page 4-38.
- Configuring aggregate routes as well as values for aggregates, such as community strings and local preference. See "Configuring Aggregate Routes" on page 4-30.
- Configuring BGP local networks. See "Configuring Local Routes (Networks)" on page 4-31.
- Configuring confederations. See "Creating a Confederation" on page 4-43.
- Configuring redistribution using route maps. See "Configuring Redistribution" on page 4-44.
- Enabling IPv6 BGP Unicast. See "Enabling/Disabling IPv6 BGP Unicast" on page 4-58.
- Configuring an IPv6 BGP Peer. See "Configuring an IPv6 BGP Peer" on page 4-58.
- Configuring the advertising of IPv4 routes for IPv6 BGP peer. See "Configuring the advertising of IPv4 routes for IPv6 peers" on page 4-67.
- Setting MD5 authentication key for IPv6 peer. See "Setting Peer Authentication" on page 4-67.
- Configuring IPv6 BGP Networks. See "Configuring IPv6 BGP Networks" on page 4-68.
- Configuring IPv6 Redistribution. See "Configuring IPv6 Redistribution" on page 4-71.
- Using policies to control BGP routing. See "Routing Policies" on page 4-78.
- Configuring GTSM for eBGP peers. see "Generalized TTL Security Mechanism (GTSM) for BGP or eBGP Peer" on page 4-89

Quick Steps for Using BGP

The following steps and points summarize enabling BGP on the OmniSwitch.

1 For BGP to be operational, the router's unique router-id and primary address must be configured. Assign the BGP local speaker's router-id and primary IP address that uniquely identifies the router in the routing domain. If these values have not been manually configured, they default to the user-defined Loopback0 interface address, if present, or to the address assigned to the first operational IP interface.

```
-> ip router router-id 1.1.1.1
-> ip router primary-address 1.1.1.1
```

2 The BGP software is not loaded automatically when the router is booted. The user must manually load the software into memory by typing the following command:

```
-> ip load bgp
```

3 Assign an Autonomous System (AS) number to the local BGP speaker. The user can change the default AS number to fit the network requirements. For example:

```
-> ip bgp autonomous-system 100
```

4 Enable the BGP protocol by entering the following command:

```
-> ip bgp admin-state enable
```

5 Create a BGP peer entry. The local BGP speaker should be able to reach this peer. The IP address assigned to the peer should be valid. For example:

```
-> ip bgp neighbor 198.45.16.145
```

6 Assign an AS number to the peer just created. All peers require an AS number. The AS number does not have to be the same as the AS number for the local BGP speaker. For example:

```
-> ip bgp neighbor 198.45.16.145 remote-as 200
```

7 As a BGP peer is not active on the network until enabled, use the following command to enable the peer created in Step 4:

```
-> ip bgp neighbor 198.45.16.145 admin-state enable
```

BGP Overview

BGP (Border Gateway Protocol) is a protocol for exchanging routing information between gateway hosts in a network of autonomous systems. BGP is the most common protocol used between gateway hosts on the Internet. The routing table exchanged between hosts contains a list of known routers, the addresses they can reach, and attributes associated with the path.

BGP is a distance vector protocol, like the Routing Information Protocol (RIP). It does not require periodic refresh of its entire routing table, but messages are sent between BGP peers to ensure a connection is active. A BGP speaker must retain the current routing table of its peers during the life of a connection.

Hosts using BGP communicate using the Transmission Control Protocol (TCP) on port 179. On connection start, BGP peers exchange complete copies of their routing tables, which can be quite large. However, only changes are exchanged after startup, which makes long running BGP sessions more efficient than shorter ones. BGP allows administrators to control routing table exchanges based on policy statements.

BGP makes it easy to use Classless Inter-Domain Routing (CIDR), which is a way to increase addresses within the network beyond the current Internet Protocol address assignment scheme. BGP's basic unit of routing information is the BGP path, which is a route to a certain set of CIDR prefixes. Paths are tagged with various path attributes, of which the most important are AS_PATH and NEXT_HOP.

One of BGP's most important functions is loop detection at the autonomous system level, using the AS_PATH attribute. The AS_PATH attribute is a list of ASs being used for data transport. The syntax of this attribute is made more complex by its need to support path aggregation, when multiple paths are collapsed into one to simplify further route advertisements. A simplified view of AS_PATH is that it is the list of Autonomous Systems that a route goes through to reach its destination. Loops are detected and avoided by checking for your own AS number in AS_PATHs received from neighboring Autonomous Systems.

An OmniSwitch using BGP could be placed at the edge of an enterprise network to handle downstream Internet traffic. An example of such a configuration would be two (2) paths to the Internet, or a dual-homed network.

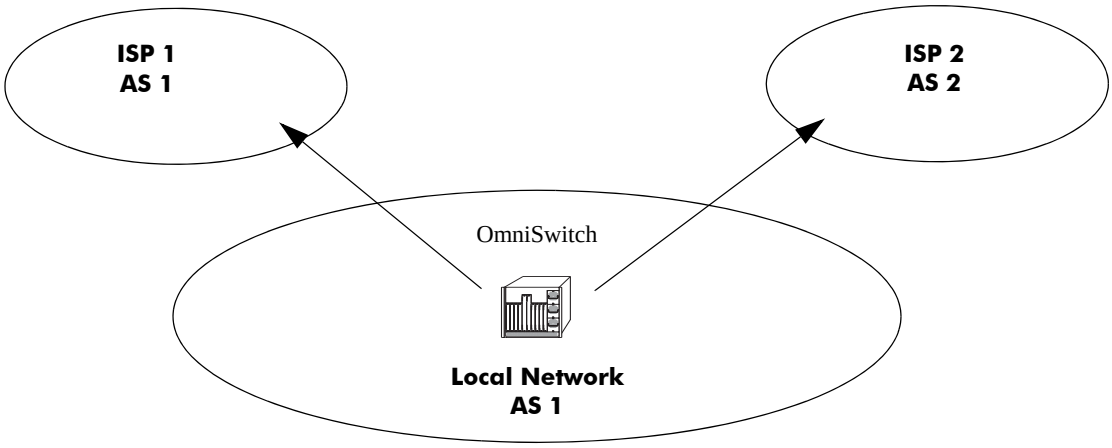


Figure 4-1 : BGP Overview

BGP is intended for use in networks with multiple autonomous systems. It is not intended to be used as an Interior Gateway protocol (IGP), such as RIP or Open Shortest Path First (OSPF). In addition, when BGP

is used as an internal routing protocol, is best used in transit autonomous systems with multiple exit points as it includes features that help routers decide among multiple exit paths.

BGP uses TCP as its transport protocol, eliminating the need for it to implement mechanisms for protocol message fragmentation, retransmission, acknowledgment, and sequencing information.

Autonomous Systems (ASs)

Exterior routing protocols were created to control the expansion of routing tables and to provide a more structured view of the Internet by segregating routing domains into separate administrations, called Autonomous Systems (ASs). Each AS has its own routing policies and unique Interior Gateway Protocols (IGP).

More specifically, an AS is a set of routers that has a single routing policy, runs under a single technical administration that commonly utilizes a single IGP (though there could be several different IGPs intermeshed to provide internal routing). To the rest of the networking world, an AS appears as a single entity.

The diagram below demonstrates the relationship of BGP and ASs:

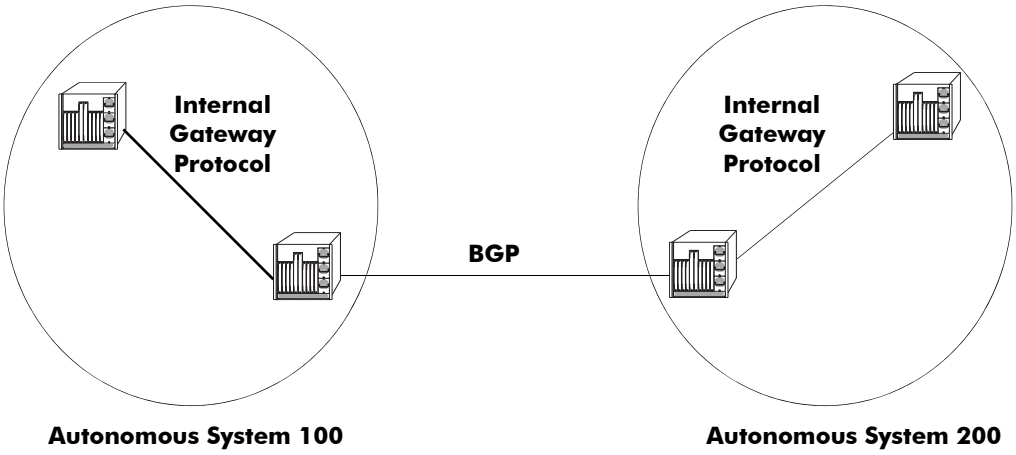


Figure 4-2 : Autonomous Systems (ASs)

Each AS has a number assigned to it by an Internet Registry, much like an IP address. BGP is the standard Exterior Gateway Protocol (EGP) used for exchanging information between ASs.

The main difference between routing within an AS (IGP) and routing outside of an AS (EGP) is that IGP policies tend to be set due to traffic concerns and technical demands, while EGP policies are set more on business relationships between corporate entities.

BGP 4-Octet Autonomous System Number (ASN)

An OmniSwitch with the BGP support for 4-octet ASN capability automatically advertises itself as being capable of handling 4-octet ASNs. OmniSwitch AOS is backward-compatible with other BGP devices that are not capable of 4-octet ASNs. The OmniSwitch supports transitive optional path attributes for interoperability with BGP devices that do not support 4-octet ASNs.This feature provides the following:

- BGP Support for 4-octet (32 bit) ASN for BGP neighbor interoperability and path attribute interoperability as per RFC 6793.

- Advertisement and discovery of 4-octet ASN capability by using the BGP Capability advertisement as specified in RFC 5492.
- Support for two new optional transitive attributes AS4_PATH and AS4_AGGREGATE. These attribute are used when new BGP speakers are interacting with OLD BGP speaker.
- To establish a neighbor relationship between non-mappable BGP 4-octet ASNs with BGP 2-octet ASNs the reserved 2-octet ASN AS_TRANS 23456 is used.
- The 4-octet AS Specific Extended Community as specified in RFC 5668 will be used with non-mappable 4-octet ASNs. If the ASN is mappable to 2-octet, the 2-octet AS specific extended community will still be used.
- The 4-octet ASN is represented in one of three ways:
  - asplain (simple decimal notation)
  - asdot+ (two 16-bit values as low-order and high-order)
  - asdot (a mixture of asplain and asdot+).

The command **ip bgp asn-format** configures the display format to be used when displaying 4-octet ASNs. This configuration changes only the output format. The input format can be in any mode when representing the ASN.

The following examples show how to configure the local BGP ASN as 65535 in the three different formats:

```
-> ip bgp autonomous-system 65535        (asplain format)
-> ip bgp autonomous-system 0.65535      (asdot+ format)
-> ip bgp autonomous-system 65535        (asdot format)
```

The following examples show how to configure the local BGP ASN as 65538 in the three different formats:

```
-> ip bgp autonomous-system 65538              (asplain format
-> ip bgp autonomous-system 1.2                (asdot+ format)
-> ip bgp autonomous-system 1.2                (asdot format)
```

The following examples show how to configure the BGP neighbor ASN as 65535 in the three different formats:

```
-> ip bgp neighbor 2.2.2.2 remote-as 65535        (asplain format)
-> ip bgp neighbor 2.2.2.2 remote-as 0.65535      (asdot+ format)
-> ip bgp neighbor 2.2.2.2 remote-as 65535        (asdot format)
```

Internal vs. External BGP

Although BGP is an exterior gateway protocol, it can still be used inside an AS as a pipe to exchange BGP updates. BGP connections inside an AS are referred to as Internal BGP (IBGP), while BGP connections between routers in separate ASs are referred to as External BGP (EBGP).

ASs with more than one connection to the outside world are called multi-homed transit ASs, and can be used to transit traffic by other ASs. Routers running IBGP are called transit routers when they carry the transit traffic through an AS.

For example, the following diagram illustrates the use of IBGP in a multihomed AS:

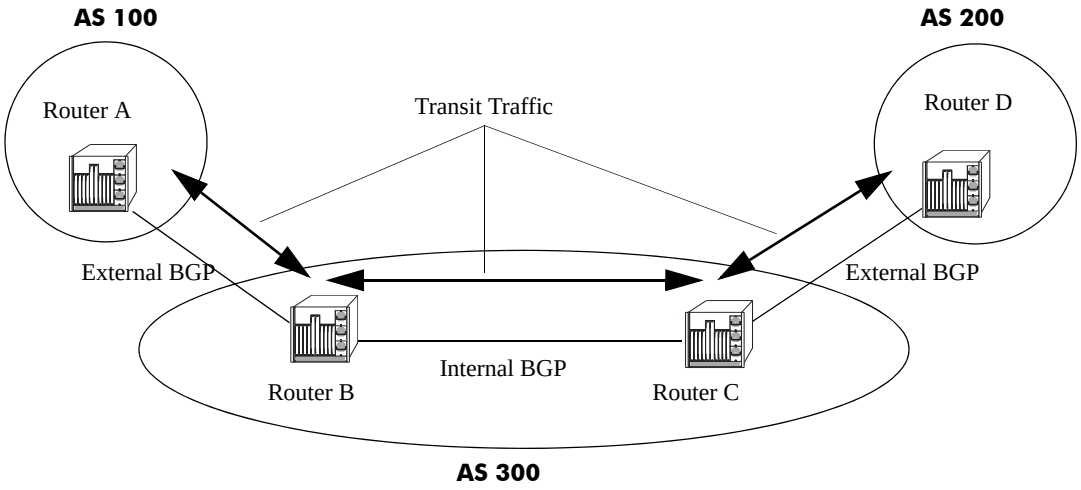


Figure 4-3 : Internal vs. External BGP

In the above diagram, AS 100 and AS 200 can send and receive traffic via AS 300. AS 300 has become a transit AS using IBGP between Router B and Router C.

Not all routers in an AS need to run BGP; in most cases, the internal routers use an IGP (such as RIP or OSPF) to manage internal AS routing. This alleviates the number of routes the internal nontransit routers must carry.

Communities

A community is a group of destinations that share some common property. A community is not restricted to one network or one autonomous system.

Communities are used to simplify routing policies by identifying routes based on a logical property rather than an IP prefix or an AS number. A BGP speaker can use this attribute in conjunction with other attributes to control which routes to accept, prefer, and pass on to other BGP neighbors.

Communities are not limited by physical boundaries, and routers in a community can belong to different ASs.

For example, a community attribute of “no export” could be added to a route, preventing it from being exported, as shown:

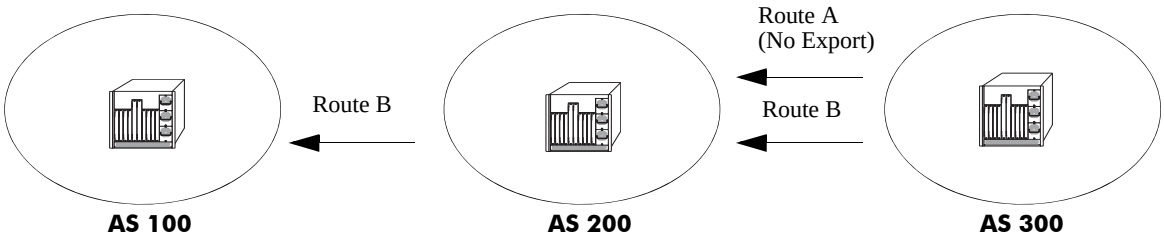


Figure 4-4 : Communities

In the above example, Route A is not propagated to AS 100 because it belongs to a community that is not to be exported by a speaker that learns it.

A route can have more than one community attribute. A BGP speaker that sees multiple community attributes in a route can act on one, several, or all of the attributes. Community attributes can be added or modified by a speaker before being passed on to other peers.

Communities are discussed further in “Working with Communities” on page 4-42.

Route Reflectors

Route reflectors are useful if the internal BGP mesh becomes very large. A route reflector is a concentration router for other BGP peers in the local network, acting as a focal point for internal BGP sessions.

Multiple client BGP routers peer with the central route server (the reflector). The router reflectors then peer with each other. Although BGP rules state that routes learned through one IBGP speaker cannot be advertised to another IBGP speaker, route reflection allows the router reflector servers to “reflect” routes, thereby relaxing the IBGP standards.

The following illustration depicts two scenarios to demonstrate the benefit of using Route Reflectors:

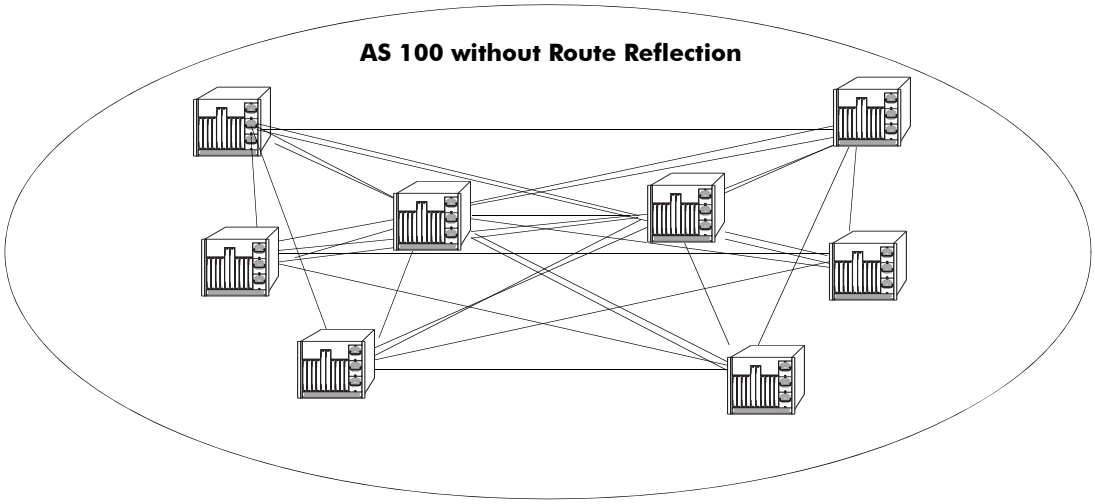


Figure 4-5 : AS 100 without Route Reflection

In the diagram below, Clients 1, 2, and 3 peer with Reflector 1, and Clients 4, 5, and 6 peer with Reflector2. Reflector 1 and 2 peer with each other. This allows each BGP speaker to maintain only one BGP session, rather than a possible seven sessions, as in the diagram above.:

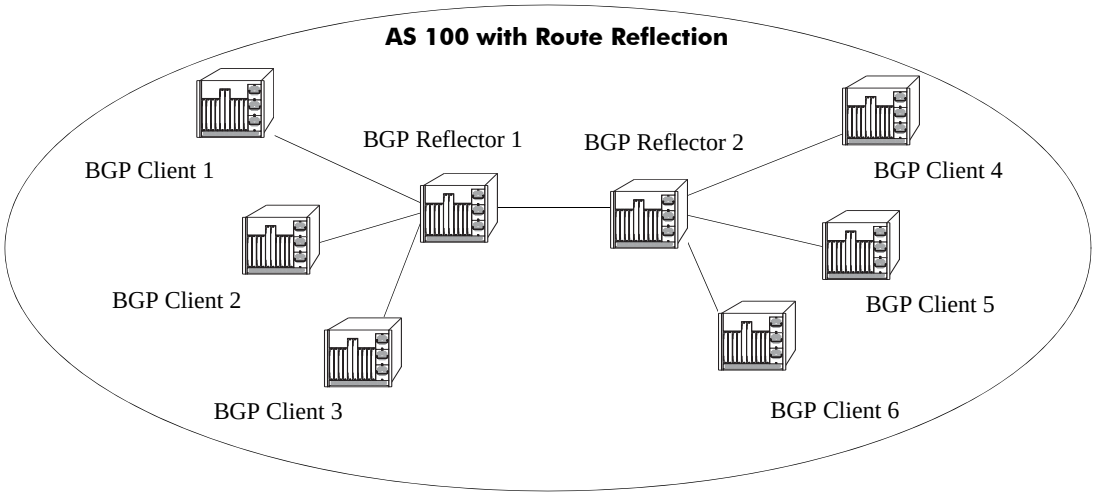


Figure 4-6 : AS 100 with Route Reflection

Since the router clients in this scenario only peer with the router reflector, the session load per router is significantly reduced. Route Reflectors are discussed further in "Setting Up Route Reflection" on page 4-38.

BGP Confederations

Confederations are another way of dealing with large networks with many BGP speakers. Like route reflectors, confederations are recommended when speakers are forced to handle large numbers of BGP sessions at the same time.

Confederations are sub ASs within a larger AS. Inside each sub AS, all the rules of IBGP apply. Since each sub AS has its own AS number, EBGP must be used to communicate between sub ASs. The following example demonstrates a simple confederation set up:

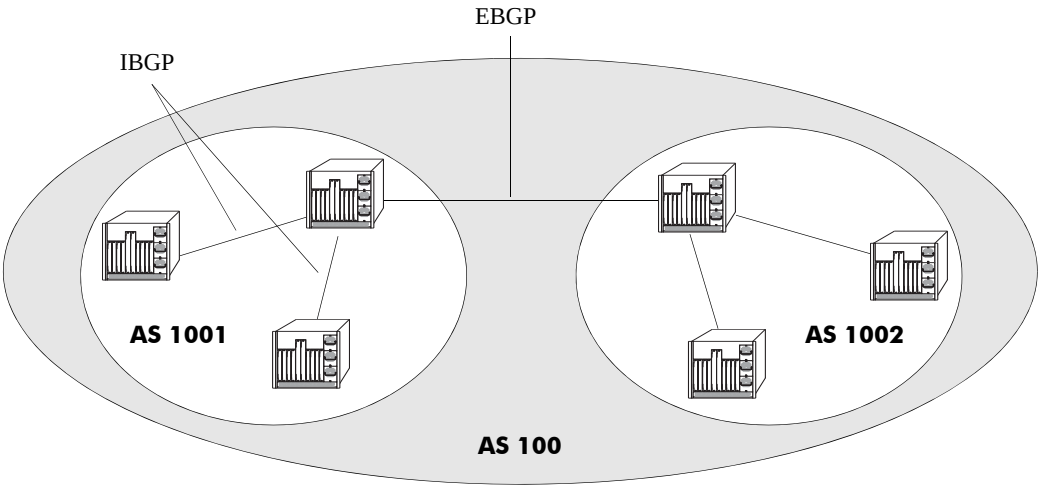


Figure 4-7 : BGP Confederations

AS 100 is now a confederation consisting of AS 1001 and AS 1002. Even though EBGP is used to communicate between AS 1001 and 1002, the entire confederation behaves as though it were using IBGP. In other words, the sub AS attributes are preserved when crossing the sub AS boundaries.

Confederations are discussed further in "Creating a Confederation" on page 4-43.

Policies

Routing policies enable route classification for importing and exporting routes. The goal of routing policies is to control traffic flow. Policies can be applied to egress and ingress traffic.

Policies act as filters to either permit or deny specified routes that are being learned or advertised from a peer. The following diagram demonstrates this concept:

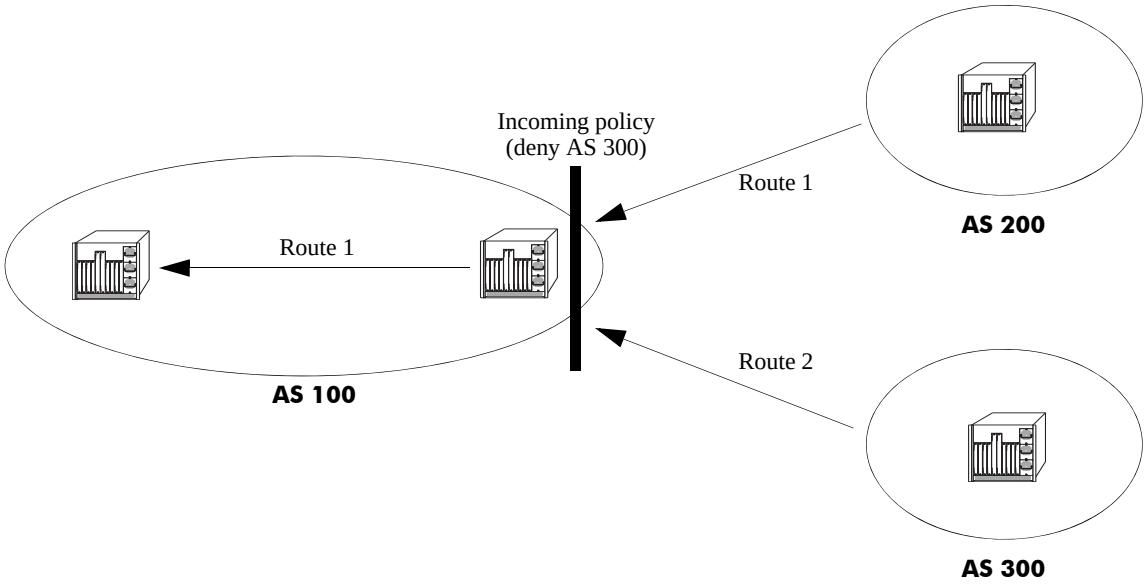


Figure 4-8 : Routing Policies

Routes from AS 200 and AS 300 are being learned by AS 100. However, there is an incoming AS Path policy at the edge of AS 100 that prevents routes that originate in AS 300 from being propagated throughout AS 100.

There are four main policy types:

- AS Path. This policy filters routes based on AS path lists. An AS path list notes all of the ASs the route travels to reach its destination.
- Community Lists. Community list policies filter routes based on the community to which a route belongs. Communities can affect route behavior based on the definition of the community.
- Prefix Lists. Prefix/Prefix6 list policies filter IPv4/IPv6 routes based on a specific network address, or a range of network addresses.
- Route Maps. Route map policies filter routes by amalgamating other policies into one policy. It is a way of combining many different filter options into one policy.

Creating and assigning policies for IPv4 and IPv6 routes is discussed in “Routing Policies” on page 4-78.

Regular Expressions

Regular expressions are used to identify AS paths for purposes of making routing decisions. In this context, an AS path is a list of one or more unsigned 16-bit AS numbers, in the range 1 through 65535.

An ordinary pattern match string looks like:

```
100 200
```

which matches any AS path containing the Autonomous System number 100 followed immediately by 200, anywhere within the AS path list. It would not match an AS path which was missing either number, or where the numbers did not occur in the correct order, or where the numbers were not adjacent to one another.

Special pattern matching characters (sometimes called metacharacters) add the ability to specify that part of the pattern must match the beginning or end of the AS path list, or that some arbitrary number of AS numbers should match, etc. The following table defines the metacharacters used in the BGP implementation.

Symbol	Description
^	Matches the beginning of the AS path list.
123	Matches the AS number 123.
.	Matches any single AS number.
?	Matches zero or one occurrence of the previous token, which must be an AS number, a dot, an alternation, or a range.
+	Matches one or more occurrences of the previous token, which must be an AS number, a dot, an alternation, or a range.
*	Matches zero or more occurrences of the previous token, which must be an AS number, a dot, an alternation, or a range.
(	Begins an alternation sequence of AS numbers. It matches any AS number listed in the alternation sequence.
|	Separates AS numbers in an alternation sequence.
)	Ends an alternation sequence of AS numbers.
[	Begin a range pair consisting of two AS numbers separated by a dash. It matches any AS number within that inclusive range.
-	Separates the endpoints of a range.
]	Ends a range pair.
$	Matches the end of the AS path list.
, _	Commas, underscores (_), and spaces are ignored.

The regular expressions configured in the router are compared against an incoming AS path list one at a time until a match is found, or until all patterns have been unsuccessfully matched. Unlike some implementations, which use a character-based pattern matching logic, the BGP implementation treats AS numbers as single tokens, providing two benefits:

- It makes writing (and reading) policies much easier.
- It enables the router to begin using the policies more quickly after startup.

For example, to identify routes originating from internal autonomous systems, use the pattern:

```
[64512-65535]$
```

which means “match any AS number from 64512 to 65535 (inclusive) which occurs at the end of the AS path.” To accomplish the same thing using character-based pattern matching, use the following pattern:

```
(_6451[2-9]_|_645[2-9][0-9]_|_64[6-9][0-9][0-9]_|_65[0-9][0-9][0-9]_)$
```

Some examples of valid regular expressions are shown in the following table:

Example		**Description**
100	Meaning:	Any route which passes through AS number 100.
	Matches:	100 200 300 300 100 100
	Doesn’t Match:	200 300
^100	Meaning:	Any routes for which the next hop is AS number 100.
	Matches:	100 200 100
	Doesn’t Match:	50 100 200
100$	Meaning:	Any route which originated from AS number 100 (AS numbers are prepended to the AS path list as they are passed on, so the originating AS is always the last number in the list).
	Matches:	100 200 200 100
	Doesn’t Match:	100 200
^100 500$	Meaning:	A route with just two hops, 100 and 500.
	Matches:	100 500
	Doesn’t Match:	100 500 600 100 200 500
100 . . 200	Meaning:	Any route with at least 4 hops, with 100 separated by any two hops from 200.
	Matches:	50 100 400 500 200 600 100 100 100 200
	Doesn’t Match:	100 200 100 100 200
(100|200).+ [500-650]$	Meaning:	Any route which begins with 100 or 200, ends with an AS number between 500 and 650 (inclusive), and is at least three hops in length. The “.+” part matches at least one (but possibly more) AS numbers.

	Matches:	100 350 501 200 250 260 270 280 600
	Doesn’t Match:	100 600 100 400 600 700
^500	Meaning:	Only routes consisting of a single AS, 500.
	Matches:	500
	Doesn’t Match:	500 600 100 500 600
[100-199]* 500 (900|950)$	Meaning:	Any route which ends with any number of occurrences of AS numbers in the range 100 to 199, followed by 500, followed by either a 900 or 950.
	Matches:	100 150 175 500 900 100 500 950
	Doesn’t Match:	100 200 500 900 100 199 500

Some examples of invalid regular expressions are shown in the following table:

Error	**Description**
66543	Number is too large. AS numbers must be in the range 1 to 65535.
64,512	Possibly an error, if the user meant the number 64512. The comma gets interpreted as a separator, thus the pattern is equivalent to the two AS numbers 64 and 512.
(100 200 | 300)	Alternation sequences must consist of single AS numbers separated by vertical bars, enclosed by parentheses.
(100*|200)	No metacharacters other than vertical bars may be included within an alteration sequence.
(100 | (200|300))	Parthentheses may not be nested. This pattern is actually equivalent to (100|200|300).
100 ^ 200	The “^” metacharacter must occur first in the pattern, as it matches the beginning of the AS path.
^500 $600	The “$” metacharacter must occur last in the pattern, as it matches the end of the AS path.
^? 100	The repetition metacharacters (?,+,*) cannot be applied to the beginning of the line. If it were legal, this pattern would be equivalent to the pattern: 100.
[1-(8|9)]*	A range cannot contain an alternation sequence.

Route Dampening

Route dampening is a mechanism for controlling route instability. If a route (or path) is frequently advertised and withdrawn, it can expend router resources. Route dampening categorizes a route as either *behaved* or *ill-behaved*. A well behaved route shows a high degree of stability over an extended period of time, while an ill-behaved route shows a high degree of instability over a short period of time. This instability is also known as *flapping*.

Route dampening can suppress (not activate) an ill-behaved route until it has achieved a certain degree of stability. Route suppression is based on the number of times a route flaps over a period of time.

Note. This mechanism does not apply to IPv6 prefixes.

The following diagram illustrates this concept:

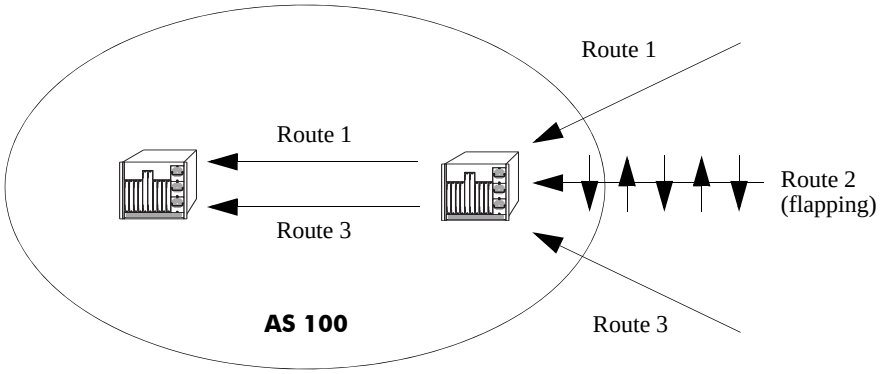


Figure 4-9 : Route Dampening

Routes 1, 2, and 3 are entering AS 100, but Route 2 (because it is flapping) has exceeded the dampening threshold. It is therefore not propagated into the AS.

The dampening threshold and suppression time of a route is determined by various factors discussed in "Controlling Route Flapping Through Route Dampening" on page 4-34.

CIDR Route Notation

Although CIDR is supported by the router, CIDR route notation is not supported on the CLI command line. For example, in order to enter the route "198.16.10.0/24" input "198.16.10.0 255.255.255.0". Some show commands, such as **ip bgp policy prefix-list** do use CIDR notation to indicate route prefixes.

BGP Configuration Overview

The following steps and points summarize configuring BGP. Not all of the following are necessary. For the necessary steps to enable BGP on the OmniSwitch, see “Quick Steps for Using BGP” on page 4-3.

1 For BGP to be operational, the router's unique router-id and primary address must be configured. Assign the BGP local speaker's router-id and primary IP address that uniquely identifies the router in the routing domain. If these values have not been manually configured, they default to the user-defined Loopback0 interface address, if present, or to the address assigned to the first operational IP interface.

```
-> ip router router-id 1.1.1.1
-> ip router primary-address 1.1.1.1
```

2 Load the BGP protocol. See “Starting BGP” on page 4-17.

3 Set up router-wide parameters, such as the router’s AS number, default local preference, and enable the BGP protocol. See “Setting Global BGP Parameters” on page 4-18.

4 Configure peers on the router. These peers may be in the same AS as the router or in a different AS. See “Configuring a BGP Peer” on page 4-24.

5 Configure peers that operate on remote routers. These peers may be in the same AS as the router or in a different AS. See “Configuring a BGP Peer” on page 4-24.

6 Configure optional parameters. There are many optional features available in the OmniSwitch implementation of BGP. These features are described in later sections of this chapter. The following is a list of BGP features that can be configured on an OmniSwitch:

- Aggregate Routes. See “Configuring Aggregate Routes” on page 4-30.
- Local networks, or routes. See “Configuring Local Routes (Networks)” on page 4-31.
- Route Dampening. See “Controlling Route Flapping Through Route Dampening” on page 4-34.
- Route Reflection. See “Setting Up Route Reflection” on page 4-38.
- Communities. See “Working with Communities” on page 4-42.
- Confederations. See “Creating a Confederation” on page 4-43.
- Policies to control BGP routing. See “Routing Policies” on page 4-78.
- Redistribution policies using route maps. See “Configuring Redistribution” on page 4-44.

Starting BGP

Before BGP is operational on the router must load it to running memory and then administratively enable the protocol using the **ip load bgp** and **ip bgp admin-state** commands. Follow these steps to start BGP.

1 Configure the router's unique router-id and primary address. Assign the BGP local speaker's router-id and primary IP address that uniquely identifies the router in the routing domain. If these values have not been manually configured, they default to the user-defined Loopback0 interface address, if present, or to the address assigned to the first operational IP interface.

```
-> ip router router-id 1.1.1.1
-> ip router primary-address 1.1.1.1
```

2 Install advanced routing image file in the active boot directory.

3 Load the BGP image into running memory by issuing the following command:

```
-> ip load bgp
```

4 Administratively enable BGP by issuing the following command:

```
-> ip bgp admin-state enable
```

Disabling BGP

The user can administratively disable BGP by issuing the following command:

```
-> ip bgp admin-state disable
```

Many BGP global commands require the user to disable the protocol before changing parameters. The following functions and commands require that the user first disable BGP before issuing them:

Parameters Requiring that BGP first be disabled

Function	Command
Router’s AS number	**ip bgp autonomous-system**
Confederation identifier	**ip bgp confederation identifier**
Default local preference	**ip bgp default local-preference**
IGP synchronization	**ip bgp synchronization**
AS Path Comparison	**ip bgp bestpath as-path ignore**
MED comparison	**ip bgp always-compare-med**
Substitute missing MED value	**ip bgp bestpath med missing-as-worst**
Equal-cost multi-path comparison	**ip bgp maximum-paths**
Route reflection	**ip bgp client-to-client reflection**
Cluster ID in route reflector group	**ip bgp cluster-id**
Fast External Fail Over	**ip bgp fast-external-failover**
Enable logging of peer changes	**ip bgp log-neighbor-changes**
Sets a confederation identification value for the local BGP speaker	**ip bgp confederation identifier**

Setting Global BGP Parameters

Many BGP parameters are applied on a router-wide basis. These parameters are referred to as *global* BGP parameters. These values are taken by BGP peers in the router unless explicitly overridden by a BGP peer command. This section describes how to enable or disable BGP global parameters.

Global BGP Defaults

Parameter Description	Command	Default Value/Comments
Enable BGP	**ip bgp admin-state**	Disabled
Router's AS number	**ip bgp autonomous-system**	1
Confederation Number	**ip bgp confederation identifier**	No confederations configured
Configure support for graceful restart on a BGP router	**ip bgp graceful-restart**	Enabled
Configure the grace period (in seconds) for a graceful BGP restart	**ip bgp graceful-restart restart-interval**	90
Configures the time interval for advertising local AS networks	**ip bgp as-origin-interval**	15 seconds
Default local preference	**ip bgp default local-preference**	100
IGP synchronization	**ip bgp synchronization**	Disabled
AS Path Comparison	**ip bgp bestpath as-path ignore**	Enabled
MED comparison on external peers	**ip bgp always-compare-med**	Disabled
Substitute missing MED value	**ip bgp bestpath med missing-as-worst**	Lowest (best) possible value
Equal-cost multi-path support	**ip bgp maximum-paths**	Disabled
Route reflection	**ip bgp client-to-client reflection**	Disabled
Cluster ID in route reflector group	**ip bgp cluster-id**	0.0.0.0
Fast External Fail Over	**ip bgp fast-external-failover**	Disabled
Enable logging of peer changes	**ip bgp log-neighbor-changes**	Disabled
Route dampening	**ip bgp dampening**	Disabled

Setting the Router AS Number

The router takes a single Autonomous System (AS) number. The user can assign one and only one AS number to a router using the **ip bgp autonomous-system** command. That same router may contain peers that belong to a different AS than the AS you assign your router. In such a case these BGP peers with a different AS would be considered external BGP (EBGP) peers and the communication with those peers would be EBGP.

The following command would assign an AS number of 14 to a router:

```
-> ip bgp autonomous-system 14
```

This command requires that you first disable the BGP protocol. If BGP were already enabled, you would actually need to issue two commands to assign the router's AS number to 14:

```
-> ip bgp admin-state disable
-> ip bgp autonomous-system 14
```

Setting the Default Local Preference

A route's local preference is an important attribute in the path selection process. In many cases, it will be the most important criteria in determining the selection of one route over another. A route obtains its local preference in one of two ways:

- By taking the default local preference established globally in the router.
- By having this default local preference manipulated by another command. The BGP peer, aggregate route, and network commands allow you to assign a local preference to a route. It is also possible to manipulate the local preference of a route through BGP policy commands.

If you want to change the default local preference value, use the **ip bgp default local-preference** command. For example, if you wanted to change the default local preference for all routes to 200, you would issue the following command:

```
-> ip bgp default local-preference 200
```

This command requires that you first disable the BGP protocol. If BGP were already enabled, you would actually need to issue two commands to change the default local preference to 200:

```
-> ip bgp admin-state disable
-> ip bgp default local-preference 200
```

Enabling AS Path Comparison

The AS path is a route attribute that shows the sequence of ASs through which a route has traveled. For example, if a path originated in AS 1, then went through AS 3, and reached its destination in AS 4, then the AS path would be:

```
4 3 1
```

A shorter AS path is preferred over a longer AS path. The AS path is always advertised in BGP route updates, however you can control whether BGP uses this attribute when comparing routes. The length of the AS path may not always indicate the effectiveness for a given route. For example, if a route has an AS path of:

```
1 3 4
```

using only 1G links, it might not be a faster path than a longer AS path of:

```
2 4 5 7
```

that uses only 10-G links.

You can disable the default state of AS comparison by specifying:

```
-> no ip bgp bestpath as-path ignore
```

This command requires that you first disable the BGP protocol. If BGP were already enabled, you would actually need to issue two commands to turn off AS path comparison:

```
-> ip bgp admin-state disable
-> no ip bgp bestpath as-path ignore
```

Controlling the use of MED Values

The Multi Exit Discriminator, or MED, is used by border routers (i.e., BGP speakers with links to neighboring autonomous systems) to help choose between multiple entry and exit points for an autonomous system. It is only relevant when an AS has more than one connection to a neighboring AS. If all other factors are equal, the path with the lowest MED value takes preference over other paths to the neighbor AS.

If received on external links, the MED may be propagated over internal links to other BGP speakers in the same AS. However, the MED is never propagated to speakers in a neighboring AS. The MED attribute indicates the weight of a particular exit point from an AS. Some exit points may be given a better MED value because they lead to higher speed connections.

The OmniSwitch implementation of BGP allows you to control MED values in the following ways:

- Compare MED values for external ASs
- Insert a MED value in routes that do not contain MEDs

The following two sections describe these MED control features.

Enabling MED Comparison for External Peers

By default, BGP only compares MEDs from peers within the same autonomous system when selecting routes. However, you can configure BGP to compare MEDs values received from external peers, or other autonomous systems. To enable MED comparison of external peers specify:

```
-> ip bgp always-compare-med
```

This command requires that you first disable the BGP protocol. If BGP were already enabled, you would actually need to issue two commands to disable MED comparison:

```
-> ip bgp admin-state disable
-> no ip bgp always-compare-med
```

Inserting Missing MED Values

A MED value may be missing in a route received from an external peer. Your can specify how a missing MED in an external BGP path is to be treated for route selection purposes. The default behavior is to treat missing MEDs as zero (best). The **ip bgp bestpath med missing-as-worst** command allows you to treat missing MEDs as 2^{32-1} (worst) for compatibility reasons.

To change the missing MED value from worst to best, enter the following command:

```
-> ip bgp bestpath med missing-as-worst
```

Synchronizing BGP and IGP Routes

In a transit-AS, BGP must ensure internal reachability to external BGP routes, prior to advertising these transit routes to external ASs. Otherwise, traffic can be lost.

The BGP rule is that a BGP router should not advertise to external neighbors destinations learned from IBGP neighbors unless those destinations are also known via an IGP. This is known as *synchronization*. If a router knows about a destination via an IGP, it is assumed that the route has already been propagated inside the AS and internal reachability is ensured.

The consequence of injecting BGP routes inside an IGP is costly. Redistributing routes from BGP into the IGP results in major overhead on the internal routers, and IGPs are really not designed to handle that many routes.

The **ip bgp synchronization** command enables or disables BGP internal synchronization. Enabling this command will force all routers (BGP and non-BGP) in an AS to learn all routes learned over external BGP. Learning the external routes forces the routing tables for all routers in an AS to be synchronized and ensure that all routes advertised within an AS are known to all routers (BGP and non-BGP). However, since routes learned over external BGP can be numerous, enabling synchronization can place an extra burden on non-BGP routers. By default, BGP internal synchronization is disabled.

To change the default state of synchronization, enter the following command:

```
-> ip bgp synchronization
```

The BGP speaker will now synchronize with the IGP.

To deactivate synchronization, enter the same command with the **no** keyword, as shown:

```
-> no ip bgp synchronization
```

Displaying Global BGP Parameters

The following list shows the commands for viewing the various aspects of BGP set with the global BGP commands:

show ip bgp	Displays the current global settings for the local BGP speaker.
show ip bgp statistics	Displays BGP global statistics, such as number of peers, active prefixes and paths.
show ip bgp aggregate-address	Displays aggregate configuration information.
show ip bgp dampening	Displays the current route dampening configuration settings.
show ip bgp dampening-stats	Displays route flap dampening statistics.
show ip bgp network	Displays information on the currently defined BGP networks.
show ip bgp path	Displays information, such as Next Hop and other BGP attributes, for every path in the BGP routing table.
show ip bgp routes	Displays information on BGP routes known to the router. This information includes whether changes to the route are in progress, whether it is part of an aggregate route, and whether it is dampened.

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

Configuring a BGP Peer

BGP supports two types of peers, or neighbors: internal and external. Internal sessions are run between BGP speakers in the same autonomous system (AS). External sessions are run between BGP peers in different autonomous systems. Internal neighbors may be located anywhere within the same autonomous system while external neighbors are adjacent to each other and share a subnet. Internal neighbors may or may not share a subnet.

BGP Peers can be configured on the OmniSwitch with the same AS number or different AS numbers. Each peer must be explicitly configured on the OmniSwitch as BGP peers and are not dynamically learned.

Note. In this document, the BGP terms “peer” and “neighbor” are used interchangeably to mean any BGP entity known to the local router.

Peer Command Defaults

The following table lists the default values for many of the peer commands:

Parameter Description	Command	Default Value/ Comments
Enables or disables BGP peer.	**ip bgp neighbor admin-state**	disabled
Assigns an AS number to the BGP peer.	**ip bgp neighbor remote-as**	1
Configures the time interval for updates between external BGP peers.	**ip bgp neighbor advertisement-interval**	30
Enables or disables BGP peer automatic restart.	**ip bgp neighbor auto-restart**	enabled
Configures this peer as a client to the local route reflector.	**ip bgp neighbor route-reflector-client**	disabled
The interval, in seconds, between BGP retries to set up a connection through the transport protocol with another peer.	**ip bgp neighbor conn-retry-interval**	120 seconds
Enables or disables BGP peer default origination.	**ip bgp neighbor default-originate**	disabled
Configures the tolerated hold time interval, in seconds, for this peer’s session, and timer interval between KEEPALIVE messages sent to this peer.	**ip bgp neighbor timers**	hold time - 90 seconds keep alive - 30 seconds
Configures the maximum number of prefixes, or paths, the local router can receive from this peer in UPDATE messages.	**ip bgp neighbor maximum-prefix**	5000

Parameter Description	Command	Default Value/ Comments
Enable or disables maximum prefix warning for a peer.	**ip bgp neighbor maximum-prefix warning-only**	80 percent
Allows external peers to communicate with each other even when they are not directly connected.	**ip bgp neighbor ebgp-multihop**	disabled
Configures the BGP peer name.	**ip bgp neighbor description**	peer IP address
Sets the BGP peer to use its own peering address as the next hop in UPDATE messages.	**ip bgp neighbor next-hop-self**	disabled
Configures the local BGP speaker to wait for this peer to establish a connection.	**ip bgp neighbor passive**	disabled
Enables or disables the stripping of private autonomous system numbers from the AS path of routes destined to this peer.	**ip bgp neighbor remove-private-as**	disabled
Enables or disables BGP peer soft reconfiguration.	**ip bgp neighbor soft-reconfiguration**	enabled
Configures this peer as a member of the same confederation as the local BGP speaker.	**ip bgp confederation neighbor**	disabled
Configures the local transport endpoint address for this neighbor's peering session.	**ip bgp neighbor update-source**	Not set until configured

Note. BGP peers are not dynamically learned. BGP peers must be explicitly configured on the router using the **ip bgp neighbor** command.

Creating a Peer

1 Create the peer and assign it an address using the **ip bgp neighbor** command. For example to create a peer with an address of 190.17.20.16, you would enter:

```
-> ip bgp neighbor 190.17.20.16
```

2 Assign an AS number to the peer using the **ip bgp neighbor remote-as** command. For example to assign the peer created in Step 1 to AS number 100, you would enter:

```
-> ip bgp neighbor 190.17.20.16 remote-as 100
```

The AS number for a peer assumes the default value, if an AS number is not configured through the **ip bgp neighbor remote-as** command.

3 You can optionally assign this peer a descriptive name using the **ip bgp neighbor description** command. Such a name may be helpful particularly in networks with connections to more than one ISP. For example, you could name peers based on their connection to a given ISP. In the example above, you could name the peer "FastISP" as follows:

```
-> ip bgp neighbor 190.17.20.16 description FastISP
```

4 Configure optional attributes for the peer. You can configure many attributes for a peer; these attributes are listed in the table below along with the commands used to configure them.

Optional BGP Peer Parameters

Peer Parameter	Command
Assigns an AS number to the BGP peer	**ip bgp neighbor remote-as**
Interval between route advertisements with external peers.	**ip bgp neighbor advertisement-interval**
Enables or disables BGP peer automatic restart.	**ip bgp neighbor auto-restart**
The interval, in seconds, between BGP retries to set up a connection via the transport protocol with another peer.	**ip bgp neighbor conn-retry-interval**
Enables or disables BGP peer default origination.	**ip bgp neighbor default-originate**
Configures the tolerated hold time interval, in seconds, for this peer's session, and timer interval between KEEPALIVE messages sent to this peer.	**ip bgp neighbor timers**
Configures the maximum number of prefixes, or paths, the local router can receive from this peer in UPDATE messages.	**ip bgp neighbor maximum-prefix**
Enable or disables maximum prefix warning for a peer.	**ip bgp neighbor maximum-prefix warning-only**
Configures the local address from which this peer will be contacted.	**ip bgp neighbor update-source**

Peer Parameter	Command
Allows external peers to communicate with each other even when they are not directly connected.	**ip bgp neighbor ebgp-multihop**
Sets the BGP peer to use next hop processing behavior.	**ip bgp neighbor next-hop-self**
Configures the local BGP speaker to wait for this peer to establish a connection.	**ip bgp neighbor passive**
Enables or disables the stripping of private autonomous system numbers from the AS path of routes destined to this peer.	**ip bgp neighbor remove-private-as**
Enables or disables BGP peer soft reconfiguration.	**ip bgp neighbor soft-reconfiguration**
Configures the check for the first AS in the ASPATH list while processing UPDATE message from BGP neighbor	**ip bgp neighbor check-first-as**

5 After entering all commands to configure a peer, you need to administratively enable the peer. The peer will not begin advertising routes until you enable it. To enable the peer in the above step, enter the **ip bgp neighbor admin-state** command:

```
-> ip bgp neighbor 190.17.20.16 admin-state enable
```

Restarting a Peer

Many BGP peer commands will automatically restart the peer once they are executed. By restarting the peer, these parameters take effect as soon as the peer comes back up. However, there are some peer commands (such as those configuring timer values) that do not reset the peer. If you want these parameters to take effect, then you must manually restart the BGP peer using the **ip bgp neighbor clear** command. The following command would restart the peer at address 190.17.20.16:

```
-> ip bgp neighbor 190.17.20.16 clear
```

The peer is not available to send or receive update or notification messages while it is restarting.

Use the **ip bgp neighbor clear soft** command to reset peer policy parameters.

Setting the Peer Auto Restart

When the auto restart is enabled, this peer will automatically attempt to restart a session with another peer after a session with that peer terminates.

To enable the auto restart feature, enter the **ip bgp neighbor auto-restart** command with the peer IP address, as shown:

```
-> ip bgp neighbor 190.17.20.16 auto-restart
```

To disable this feature, enter the following:

```
-> no ip bgp neighbor 190.17.20.16 auto-restart
```

Changing the Local Router Address for a Peer Session

By default, TCP connections to a peer's address are assigned to the closest interface based on reachability. Any operational local interface can be assigned to the BGP peering session by explicitly forcing the TCP connection to use the specified interface. The **ip bgp neighbor update-source** command sets the local interface address or the name through which this BGP peer can be contacted.

For example, to configure a peer with an IP address of 120.5.4.6 to be contacted via 120.5.4.10, enter the **ip bgp neighbor update-source** command as shown:

```
-> ip bgp neighbor 120.5.4.6 update-source 12.5.4.10
```

Alternatively, you can enter the name of the local IP interface, instead of the IP address as shown below:

```
-> ip bgp neighbor 120.5.4.6 update-source vlan-23
```

Clearing Statistics for a Peer

BGP tracks the number of messages sent to and received from other peers. It also breaks down messages into UPDATE, NOTIFICATION, and TRANSITION categories. You can reset, or clear, the statistics for a peer using the **ip bgp neighbor stats-clear** command. For example the following use of the **ip bgp neighbor stats-clear** command would clear statistics for the peer at address 190.17.20.16:

```
-> ip bgp neighbor 190.17.20.16 stats-clear
```

The statistics that are cleared are shown in the **show ip bgp neighbors statistics** command. The following is an example of output from this command:

```
-> show ip bgp neighbors statistics 190.17.20.16

Neighbor address                    = 190.17.20.16,
# of UP transitions                 = 0,
Time of last UP transition          = 00h:00m:00s,
# of DOWN transitions               = 0,
Time of last DOWN transition        = 00h:00m:00s,
Last DOWN reason                    = none,
# of msgs rcvd                      = 0,
# of Update msgs rcvd               = 0,
# of prefixes rcvd                  = 0,
# of Route Refresh msgs rcvd        = 0,
# of Notification msgs rcvd         = 0,
Last rcvd Notification reason       = none [none]
Time last msg was rcvd              = 00h:00m:00s,
# of msgs sent                      = 0,
# of Update msgs sent               = 0,
# of Route Refresh msgs sent        = 0
# of Notification msgs sent         = 0,
Last sent Notification reason       = none [none],
Time last msg was sent              = 00h:00m:00s
```

Setting Peer Authentication

You can set which MD5 authentication key this router will use when contacting a peer. To set the MD5 authentication key, enter the peer IP address and key with the **ip bgp neighbor md5 key** command:

```
-> ip bgp neighbor 123.24.5.6 md5 key keyname
```

The peer with IP address 123.24.5.6 will be sent messages using “keyname” as the encryption password. If this is not the password set on peer 123.24.5.6, then the local router will not be able to communicate with this peer.

Configuring the advertising of IPv4 routes for an IP BGP peer

You can enable or disable the advertising of IPv4 routes on an IPv4 neighbor. To enable the advertisement of IPv4 unicast capability to the IPv4 BGP peer, use the following command:

```
-> ip bgp neighbor 172.22.2.115 activate-ipv4
```

The advertising capability is enabled for the BGP peer with IP address 172.22.2.115.

Note. The advertisement of IPv4 unicast capability is enabled by default.

To disable the advertising capability use the no form of the command.

```
-> no ip bgp neighbor 172.22.2.115 activate-ipv4
```

Setting the Peer Route Advertisement Interval

The route advertisement interval specifies the frequency at which routes external to the autonomous system are advertised. These advertisements are also referred to as UPDATE messages. This interval applies to advertisements to external peers.

To set the advertisement interval, enter the number of seconds in conjunction with the **ip bgp neighbor advertisement-interval** command, as shown:

```
-> ip bgp neighbor 123.24.5.6 advertisement-interval 50
```

The interval is now set to 50 seconds.

Configuring a BGP Peer with the Loopback0 Interface

Loopback0 is the name assigned to an IP interface to identify a consistent address for network management purposes. The Loopback0 interface is not bound to any VLAN, so it will always remain operationally active. This differs from other IP interfaces in that if there are no active ports in the VLAN, all IP interface associated with that VLAN are not active. In addition, the Loopback0 interface provides a unique IP address for the switch that is easily identifiable to network management applications.

It is possible to create BGP peers using the Loopback0 IP interface address of the peering router and binding the source (i.e., outgoing IP interface for the TCP connection) to its own configured Loopback0 interface. The Loopback0 IP interface address can be used for both Internal and External BGP peer sessions. For EBGP sessions, if the External peer router is multiple hops away, the **ebgp-multihop** parameter may need to be used.

The following example configures a BGP peering session using a Loopback0 IP interface address:

```
-> ip bgp neighbor 2.2.2.2 update-source Loopback0
```

See the *OmniSwitch AOS Release 8 Network Configuration Guide* for more information about configuring an IP Loopback0 interface.

Configuring Aggregate Routes

Aggregate routes are used to reduce the size of routing tables by combining the attributes of several different routes and allowing a single aggregate route to be advertised to peers.

You cannot aggregate an address (for example, 100.10.0.0) if you do not have at least one more-specific route of the address (for example, 100.10.20.0) in the BGP routing table.

Aggregate routes do not need to be known to the local BGP speaker.

1 Indicate the address and mask for the aggregate route using the **ip bgp aggregate-address** command:

```
-> ip bgp aggregate-address 172.22.2.0 255.255.255.0
```

2 Suppress the individual routes in the 172.22.2.0 network and advertise only one route using the **ip bgp aggregate-address** command with the **summary-only** parameter:

```
-> ip bgp aggregate-address 172.22.2.0 255.255.255.0 summary-only
```

3 Optional. When an aggregate route is created BGP does not aggregate the AS paths of all routes included in the aggregate. However, you may specify that a new AS path be created for the aggregate route that includes the ASs traversed for all routes in the aggregate. To specify that the AS path also be aggregated use the **ip bgp aggregate-address as-set** command. For example:

```
-> ip bgp aggregate-address 172.22.2.0 255.255.255.0 as-set
```

4 Optional. By default an aggregate route suppresses the advertisement of all more-specific routes within the aggregate. This suppression of routes is the function of an aggregate route. However, you can disable route summarization through the no **ip bgp aggregate-address summary-only**. For example:

```
-> no ip bgp aggregate-address 172.22.2.0 255.255.255.0 summary-only
```

5 Optional. You can manipulate several BGP attributes for routes included in this aggregate route. These attributes and the corresponding commands used to manipulate them are shown in the table below:

Optional Aggregate Route Attribute Manipulation

BGP Attribute	Command
Community list for this aggregate route	**ip bgp aggregate-address community**
Local preference value for this aggregate. This value overrides the value set in the **ip bgp default-lpref** command.	**ip bgp aggregate-address local-preference**
MED value for this aggregate route.	**ip bgp aggregate-address metric**

6 Once you have finished configuring values for this aggregate route, enable it using the **ip bgp aggregate-address admin-state** command. For example:

```
-> ip bgp aggregate-address 172.22.2.0 255.255.255.0 admin-state enable
```

Configuring Local Routes (Networks)

A local BGP network is used to indicate to BGP that a network should originate from a specified router. A network must be known to the local BGP speaker; it also must originate from the local BGP speaker.

Networks have some parameters that can be configured, such as **local-preference**, **community**, and **metric**.

Adding the Network

To add a local network to a BGP speaker, use the IP address and mask of the local network in conjunction with the **ip bgp network** command, as shown:

```
-> ip bgp network 172.20.2.0 255.255.255.0
```

In this example, network 172.20.2.0 with a mask of 255.255.255.0 is the local network for this BGP speaker.

To remove the same network from the speaker, enter the same command with the no keyword, as shown:

```
-> no ip bgp network 172.20.2.0 255.255.255.0
```

The network would now no longer be associated as the local network for this BGP speaker.

Enabling the Network

Once the network has been added to the speaker, it must be enabled on the speaker. To do this, enter the IP address and mask of the local network in conjunction with the **ip bgp network admin-state** command, as shown:

```
-> ip bgp network 172.20.2.0 255.255.255.0 admin-state enable
```

In this example, network 172.20.2.0 with a mask of 255.255.255.0 has now been enabled.

To disable the same network, enter the following:

```
-> ip bgp network 172.20.2.0 255.255.255.0 admin-state disable
```

The network would now be disabled, though not removed from the speaker.

Configuring Network Parameters

Once a local network is added to a speaker, you can configure three parameters that are attached to routes generated by the **ip bgp network** command. These three attributes are the local preference, the community, and the route metric.

Local Preference

The local preference is a degree of preference to be given to a specific route when there are multiple routes to the same destination. The higher the number, the higher the preference. For example, a route with a local preference of 50 will be used before a route with a local preference of 30.

To set the local preference for the local network, enter the IP address and mask of the local network in conjunction with the **ip bgp network local-preference** command and value, as shown:

```
-> ip bgp network 172.20.2.0 255.255.255.0 local-preference 600
```

The local preference for routes generated by the network is now 600.

Community

Communities are a way of grouping BGP destination addresses that share some common property. Adding the local network to a specific community indicates that the network shares a common set of properties with the rest of the community.

To add a network to a community, enter the local network IP address and mask in conjunction with the **ip bgp network community** command and name, as shown:

```
-> ip bgp network 172.20.2.0 255.255.255.0 community 100:200
```

Network 172.20.2.0, mask 255.255.255.0, is now in the 100:200 community.

To remove the local network from the community, enter the local network as above with the community set to “none”, as shown:

```
-> ip bgp network 172.20.2.0 255.255.255.0 community none
```

The network is now no longer in any community.

Metric

A metric for a network is the Multi-Exit Discriminator (MED) value. This value is used when announcing this network to internal peers; it indicates the best exit point from the AS, assuming there is more than one. A lower value indicates a more preferred exit point. For example, a route with a MED of 10 is more likely to be used than a route with an MED of 100.

To set the network metric value, enter the network IP address and mask in conjunction with the **ip bgp network metric** command and value, as shown:

```
-> ip bgp network 172.20.2.0 255.255.255.0 metric 100
```

Network 172.20.2.0, mask 255.255.255.0, is now set with a metric of 100.

Viewing Network Settings

To view the network settings for all networks assigned to the speaker, enter the **show ip bgp network** command, as shown:

```
-> show ip bgp network
```

A display similar to the following appears:

```
Network         Mask            Admin state Oper state
----------------+---------------+-----------+----------
155.132.40.0    255.255.255.0   disabled    not_active
155.132.1.3     255.255.255.255 disabled    not_active
```

To display a specific network, enter the same command with the network IP address and mask. For example,:

```
-> show ip bgp network 172.20.2.0 255.255.255.0
```

A display similar to the following appears:

```
Network address          = 172.20.2.0,
Network mask             = 255.255.255.0,
Network admin state      = disabled,
Network oper state       = not_active,
Network metric           = 0,
Network local pref       = 0,
Network community string = 0:500 400:1 300:2
```

Controlling Route Flapping Through Route Dampening

Route dampening minimizes the effect of flapping routes in a BGP network. Route flapping occurs when route information is updated erratically, such as when a route is announced and withdrawn at a rapid rate. Route flapping can cause problems in networks connected to the Internet, where route flapping will involve the propagation of many routes. Route dampening suppresses flapping routes and designates them as unreachable until they flap at a lower rate.

You can configure route dampening to adapt to the frequency and duration of a particular route that is flapping. The more a route flaps during a period of time, the longer it will be suppressed.

Each time a route flaps (i.e., withdrawn from the routing table), its “instability metric” is increased by 1. Once a route’s instability metric reaches the *suppress value*, it is suppressed and no longer advertised. The instability metric may continue to increase even after the route is suppressed.

A route’s instability metric may be reduced. It is reduced once the route stops flapping for a given period of time. This period of time is referred to as the *half-life duration*. If a suppressed route does not flap for a given half-life duration, then its instability metric will be cut in half. As long as the route continues to be stable, its instability metric will be reduced until it reaches the *reuse value*. Once below the reuse value, a route will be re-advertised.

Example: Flapping Route Suppressed, then Unsuppressed

Consider, for example, a route that has started to flap. Once this route starts exhibiting erratic behavior, BGP begins tracking the instability metric for the route. This particular route flaps more than 300 times, surpassing the cutoff value of 300. BGP stops advertising the route; the route is now suppressed. The route continues to flap and its instability metric reaches 1600.

Now the route stops flapping. In fact, it does not flap for 5 minutes, which is also the half-life duration defined for BGP routes. The instability metric is reduced to 800. The route remains stable for another 5 minutes and the instability metric is reduced to 400. After another 5 minutes of stability, the route’s instability metric is reduced to 200, which is also the defined reuse value. Since the instability metric for the route has dropped below the reuse value, BGP will begin re-advertising it again.

The following chart illustrates what happens to the described route in the above scenario:

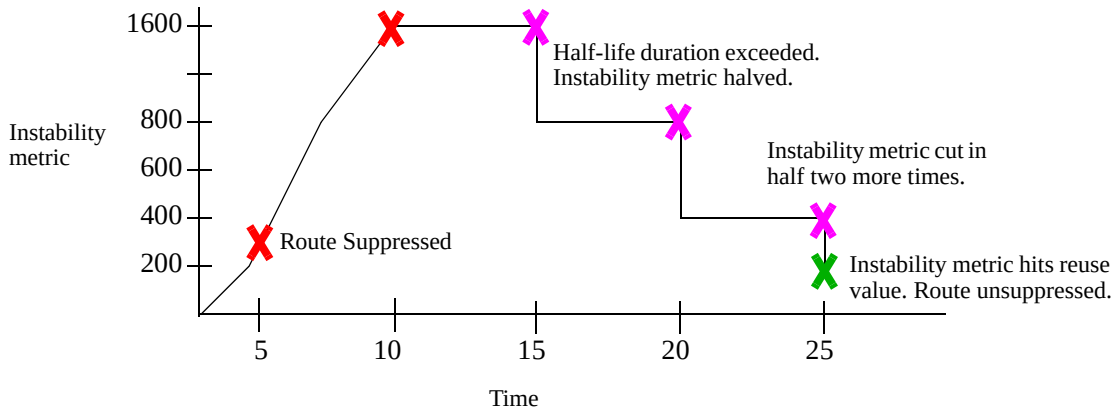


Figure 4-10 : Flapping Route Suppressed, then Unsuppressed

Enabling Route Dampening

Route dampening must be enabled before it effects routes. To enable route dampening on a BGP router, enter the **ip bgp dampening** command, as shown:

```
-> ip bgp dampening
```

To disable route dampening, enter the following:

```
-> no ip bgp dampening
```

Configuring Dampening Parameters

There are several factors in configuring route dampening. These factors work together to determine if a route should be dampened, and for how long. The values all have defaults that are in place when dampening is enabled. It is possible to change these values, using the **ip bgp dampening** command with variables. The variables for these parameters must be entered together, in one command, in order. This is demonstrated in the following sections.

- Setting the Reach Halflife. The reach halflife is the number of seconds a route can be reached, without flapping, before the penalty number (of flaps) is reduced by half. See "Setting the Reach Halflife" on page 4-35 for instructions on how this is done.
- Setting the Reuse Value. The reuse value determines if a route is advertised again. See "Setting the Reuse Value" on page 4-36 for instructions on how this is done.
- Setting the Suppress Value. The suppress value is the number of route withdrawals required before the route is suppressed. See "Setting the Suppress Value" on page 4-36 for instructions on how this is done.
- Setting the Maximum Suppress Holdtime. The maximum holdtime is the number of seconds a route stays suppressed. See "Setting the Maximum Suppress Holdtime" on page 4-36 for instructions on how this is done.

Setting the Reach Halflife

The reach halflife value is the number of seconds that pass before a route is re-evaluated in terms of flapping. After the number of seconds set for halflife has passed, and a route has not flapped, then its total flap count is reduced by half.

For example, if the reach halflife is set at 500 seconds, and a reachable route with a flap count of 300 does not flap during this time, then its flap count is reduced to 150.

To change one variable to a number different than its default value, you must enter all of the variables with the **ip bgp dampening** command in the correct order.

For example, to set the reach halflife value to 500, enter the halflife value and other variables with the following command, as shown:

```
-> ip bgp dampening half-life 500 reuse 200 suppress 300 max-suppress-time 1800
```

In this example, the other variables have been set to their default values. The reach halflife is now set to 500. The default values for the reach halflife is 300.

Setting the Reuse Value

The dampening reuse value is used to determine if a route should be re-advertised. If the number of flaps for a route falls below this number, then the route is re-advertised. For example, if the reuse value is set at 150, and a route with 250 flaps exceeds the reach halflife it would be re-advertised as its flap number would now be 125.

To change one variable to a number different than its default value, you must enter all of the variables with the **ip bgp dampening** command in the correct order.

For example, to set the reuse value to 500, enter the reuse value and other variables with the following command, as shown:

```
-> ip bgp dampening half-life 300 reuse 500 suppress 300 max-suppress-time 1800
```

In this example, the other variables have been set to their default values. The reuse value is now set to 500. The default value is 200.

Setting the Suppress Value

The dampening suppress value sets the number of times a route can flap before it is suppressed. A suppressed route is not advertised. For example, if the cutoff value is set at 200, and a route flaps 201 times, it will be suppressed.

To change one variable to a number different than its default value, you must enter all of the variables with the **ip bgp dampening** command in the correct order.

For example, to set the suppress value to 500, enter the suppress value and other variables with the following command, as shown:

```
-> ip bgp dampening half-life 300 reuse 200 suppress 500 max-suppress-time 1800
```

In this example, the other variables have been set to their default values. The suppress value is now set to 500. The default value is 300.

Setting the Maximum Suppress Holdtime

The maximum suppress holdtime is the number of seconds a route stays suppressed once it has crossed the dampening cutoff flapping number. For example, if the maximum holdtime is set to 500, once a route is suppressed the local BGP speaker would wait 500 seconds before advertising the route again.

To change one variable to a number different than its default value, you must enter all of the variables with the **ip bgp dampening** command in the correct order.

For example, to set the maximum suppress holdtime value to 500, enter the maximum suppress holdtime value and other variables with the following command, as shown:

```
-> ip bgp dampening half-life 300 reuse 200 suppress 300 max-suppress-time 500
```

In this example, the other variables have been set to their default values. The maximum suppress holdtime is now set to 500 seconds. The default value is 1800 seconds.

Clearing the History

By clearing the dampening history, you are resetting all of the dampening information on all of the routes back to zero, as if dampening had just been activated. Route flap counters are reset and any routes that were suppressed due to route flapping violations are unsuppressed. Dampening information on the route will start re-accumulating as soon as the command is entered and the statistics are cleared.

To clear the dampening history, enter the following command:

```
-> ip bgp dampening clear
```

Displaying Dampening Settings and Statistics

To display the current settings for route dampening, enter the following command:

```
-> show ip bgp dampening
```

A display similar to the following will appear:

```
Admin Status                    = disabled,
Half life value (seconds)       = 300,
Reuse value (seconds)           = 200
Suppress time (seconds)         = 300,
Max suppress time (seconds)     = 1800,
```

To display current route dampening statistics, enter the following command:

```
-> show ip bgp dampening-stats
```

A display similar to the following will appear:

```
Network          Mask             From             Flaps Duration    FOM
---------------+---------------+---------------+-----+-----------+-----
155.132.44.73   255.255.255.255 192.40.4.121    8     00h:00m:35s 175
```

Setting Up Route Reflection

BGP requires that all speakers in an autonomous system be fully meshed (i.e., each speaker must have a peer connection to every other speaker in the AS) so that external routing information can be distributed to all BGP speakers in an AS. However, fully meshed configurations are difficult to scale in large networks. For this reason, BGP supports *route reflection*, a configuration in which one or more speakers—route reflectors—handle intra-AS communication among all BGP speakers.

In a fully meshed BGP configuration, a BGP speaker that receives an external route must re-advertise the route to all internal peers. In the illustration below, BGP speaker A receives a route from an external BGP speaker and advertises it to both Speakers B and C in its autonomous system. Speakers B and C do not re-advertise the route to each other so as to prevent a routing information loop.

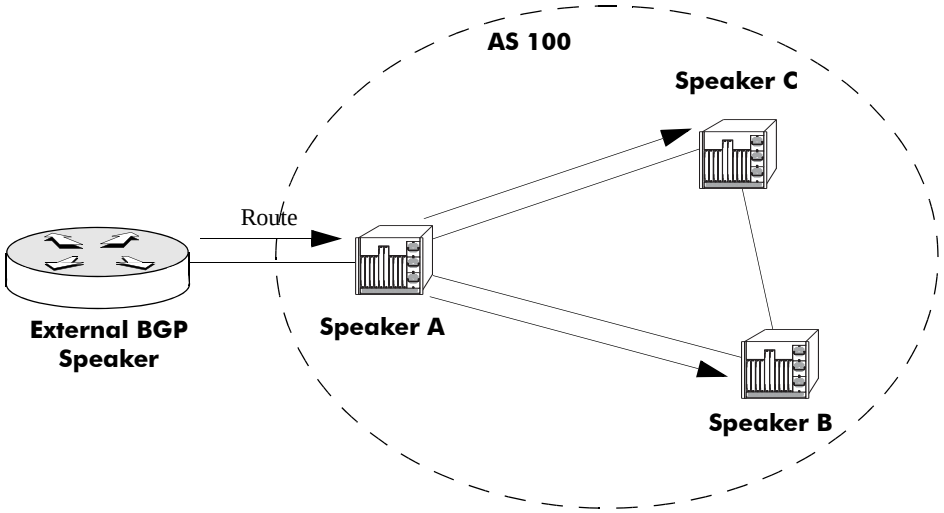


Figure 4-11 : Fully Meshed BGP Peers

In the above example, Speakers B and C do not re-advertise the external route they each received from Speaker A. However, this fundamental routing rule is relaxed for BGP speakers that are route reflectors.

This same configuration using a route reflector would not require that all BGP speakers be fully meshed. One of the speakers is configured to be a route reflector for the group. In this case, the route reflector is Speaker C. When the route reflector (Speaker C) receives route information from Speaker A it advertises the information to Speaker B. This set up eliminates the peer connection between Speakers A and B.

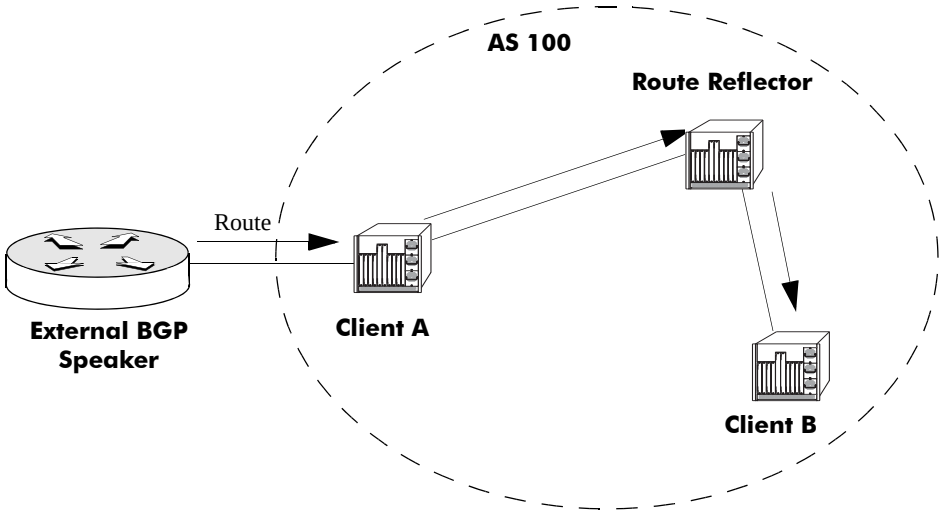


Figure 4-12 : Partly Meshed BGP Peers

The internal peers of a route reflector are divided into two groups: client peers and non-client peers. The route reflector sits between these two groups and reflects routes between them. The route reflector, its *clients*, and *non-clients* are all in the same autonomous system.

The route reflector and its clients form a *cluster*. The client peers do not need to be fully meshed (and therefore take full advantage of route reflection), but the non-client peers must be fully meshed. The following illustration shows a route reflector, its clients within a cluster, and its non-client speakers outside the cluster.

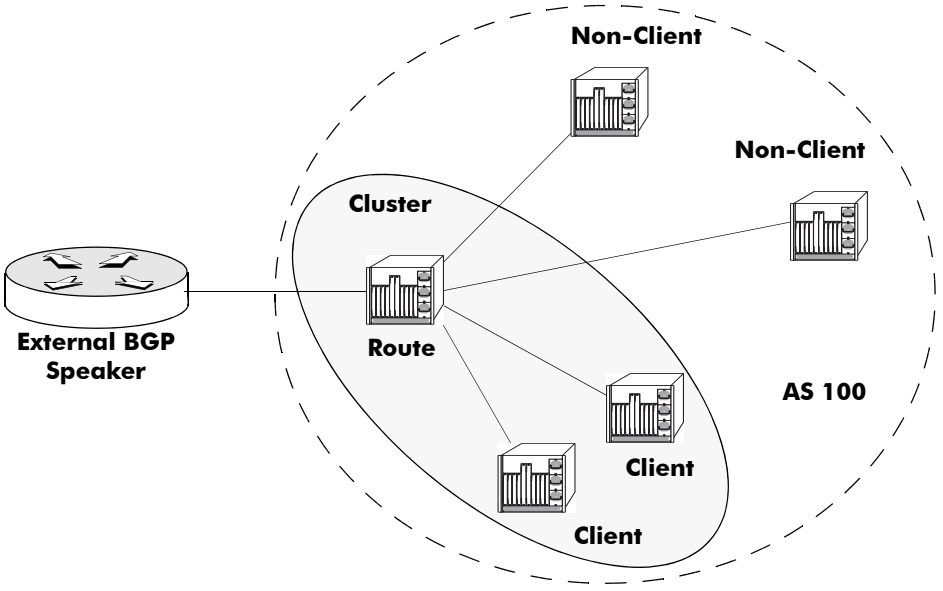


Figure 4-13 : Route Reflector, Clients, and Non-Clients

Note that the non-client BGP speakers are fully meshed with each other and that the client speakers in the cluster do not communicate with the non-client speakers.

When a route reflector receives a route it, selects the best path based on its policy decision criteria. The internal peers to which the route reflector advertises depends on the source of the route. The table below shows the rules the reflector follows when advertising path information:

Route Received From...	Route Advertised To...
External BGP Router	All Clients and Non-Clients
Non-Client Peer	All Clients
Client Peer	All Clients and Non-Clients

Configuring Route Reflection

1 Disable the BGP protocol by specifying:

```
-> ip bgp admin-state disable
```

2 Specify this router as a route reflector, using the **ip bgp client-to-client reflection** command:

```
-> ip bgp client-to-client reflection
```

The route reflector will follow the standard rules for client route advertisement (i.e., routes from a client are sent to all clients and non-clients, except the source client).

3 Indicate the client peers for this route reflector. For all internal peers (same AS as the router) that are to be clients specify the **ip bgp neighbor route-reflector-client** command, for IPv4 peers and **ipv6 bgp neighbor route-reflector-client** for IPv6 peers.

For example,

If you wanted the peer at IPv4 address 190.17.20.16 to become a client to the local BGP route-reflector, then you would specify the following command:

```
-> ip bgp neighbor 190.17.20.16 route-reflector-client
```

If you wanted the peer at IPv6 address 2001::1 to become a client to the local BGP route-reflector, then you would specify the following command:

```
-> ipv6 bgp neighbor 2001::1 route-reflector-client
```

4 Repeat Step 3 for all internal peers that are to be clients of the route reflector.

Redundant Route Reflectors

A single BGP speaker will usually act as the reflector for a cluster of clients. In such a case, the cluster is identified by the router ID of the reflector. It is possible to add redundancy to a cluster by configuring more than one route reflector, eliminating the single point of failure. Redundant route reflectors must be identified by a 4-byte cluster ID, which is specified in the **ip bgp cluster-id** command. All route reflectors in the same cluster must be fully meshed and should have the exact same client and non-client peers.

Note. Using many redundant reflectors is not recommended as it places demands on the memory required to store routes for all redundant reflectors' peers.

To configure a redundant route reflector for this router, use the **ip bgp cluster-id** command. For example to set up a redundant route reflector at 190.17.21.16, you would enter:

```
-> ip bgp cluster-id 190.17.21.16
```

Working with Communities

Distribution of routing information in BGP is typically based on IP address prefixes or on the value of the AS_PATH attributes. To facilitate and simplify the control of routing information, destinations can be grouped into communities and routing decisions can be applied based on these communities.

Communities are identified by using the numbering convention of the AS and the community number, separated by a colon (for example, 200:500)

There are a few well known communities defined (in RFC 1997) that do not require the numbering convention. Their community numbers are reserved and thus can be identified by name only. These are listed below:

- **no-export**. Routes in this community are advertised within the AS but not beyond the local AS.
- **no-advertise**. Routes in this community are not advertised to any peer.
- **no-export-subconfed**. Routes in this community are not advertised to any external BGP peer.

Communities are added to routes using the policy commands, as described in “Routing Policies” on page 4-78.

Creating a Confederation

A confederation is a grouping of ASs that together form a super AS. To BGP external peers, a confederation appears as another AS even though the confederation has multiple ASs within it. Within a confederation ASs can distinguish among one another and will advertise routes using EBGP.

1 Specify the confederation identifier for the local BGP router. This value is used to identify the confederation affiliation of routes in advertisements. This value is essentially an AS number. To assign a confederation number to the router use the **ip bgp confederation identifier** command. For example, to assign a confederation value of 2, you would enter:

```
-> ip bgp confederation-identifier 2
```

2 Indicate whether a peer belongs to the confederation configured on this router using the **ip bgp confederation neighbor** command. For example to assign the peer at 190.17.20.16 to confederation 2, you would enter:

```
-> ip bgp confederation neighbor 190.17.20.16
```

To configure the IPv6 BGP peer as a member of the same confederation as the local BGP speaker, use the **ip bgp confederation neighbor6** command. For example, to assign the peer at 2001::1 to confederation 2, you would enter:

```
-> ip bgp confederation neighbor6 2001::1
```

3 Repeat Step 2 for all peers that need to be assigned to the confederation.

Configuring Redistribution

It is possible to configure the BGP protocol to advertise routes learned from other routing protocols (external routes) into the BGP network. Such a process is referred to as route redistribution and is configured using the **ip redist** command.

BGP redistribution uses route maps to control how external routes are learned and distributed. A route map consists of one or more user-defined statements that can determine which routes are allowed or denied access to the BGP network. In addition a route map may also contain statements that modify route parameters before they are redistributed.

When a route map is created, it is given a name to identify the group of statements that it represents. This name is required by the **ip redist** command. Therefore, configuring BGP route redistribution involves the following steps:

1 Create a route map, as described in "Using Route Maps" on page 4-44.

2 Configure redistribution to apply a route map, as described in "Configuring Route Map Redistribution" on page 4-48.

Using Route Maps

A route map specifies the criteria that are used to control redistribution of routes between protocols. Such criteria is defined by configuring route map statements. There are three different types of statements:

- **Action**. An action statement configures the route map name, sequence number, and whether or not redistribution is permitted or denied based on route map criteria.
- **Match.** A match statement specifies criteria that a route must match. When a match occurs, then the action statement is applied to the route.
- **Set.** A set statement is used to modify route information before the route is redistributed into the receiving protocol. This statement is only applied if all the criteria of the route map is met and the action permits redistribution.

The **ip route-map** command is used to configure route map statements and provides the following **action**, **match**, and **set** parameters:

ip route-map action ...	ip route-map match ...	ip route-map set ...
permit **deny**	**ip-address** **ip-nexthop** **ipv6-address** **ipv6-nexthop** **tag** **ipv4-interface** **ipv6-interface** **metric** **route-type**	**metric** **metric-type** **tag** **community** **local-preference** **level** **ip-nexthop** **ipv6-nexthop**

Refer to the "IP Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide* for more information about the "ip route-map" command parameters and usage guidelines.

Once a route map is created, it is then applied using the **ip redist** command. See "Configuring Route Map Redistribution" on page 4-48 for more information.

Creating a Route Map

When a route map is created, it is given a name (up to 20 characters), a sequence number, and an action (permit or deny). Specifying a sequence number is optional. If a value is not configured, then the default value is used.

To create a route map, use the **ip route-map** command with the **action** parameter. For example,

```
-> ip route-map ospf-to-bgp sequence-number 10 action permit
```

The above command creates the ospf-to-bgp route map, assigns a **sequence number** of 10 to the route map, and specifies a **permit** action.

To optionally filter routes before redistribution, use the **ip route-map** command with a **match** parameter to configure match criteria for incoming routes. For example,

```
-> ip route-map ospf-to-bgp sequence-number 10 match tag 8
```

The above command configures a match statement for the ospf-to-bgp route map to filter routes based on their tag value. When this route map is applied, only OSPF routes with a tag value of eight are redistributed into the BGP network. All other routes with a different tag value are dropped.

Note. Configuring match statements is not required. However, if a route map does not contain any match statements and the route map is applied using the **ip redist** command, the router redistributes *all* routes into the network of the receiving protocol.

To modify route information before it is redistributed, use the **ip route-map** command with a **set** parameter. For example,

```
-> ip route-map ospf-to-bgp sequence-number 10 set tag 5
```

The above command configures a set statement for the ospf-to-bgp route map that changes the route tag value to five. Because this statement is part of the ospf-to-bgp route map, it is only applied to routes that have an existing tag value equal to eight.

The following is a summary of the commands used in the above examples:

```
-> ip route-map ospf-to-bgp sequence-number 10 action permit
-> ip route-map ospf-to-bgp sequence-number 10 match tag 8
-> ip route-map ospf-to-bgp sequence-number 10 set tag 5
```

To verify a route map configuration, use the **show ip route-map** command:

```
-> show ip route-map
Route Maps: configured: 1 max: 200
Route Map: ospf-to-bgp Sequence Number: 10 Action permit
  match tag 8
  set tag 5
```

Deleting a Route Map

Use the **no** form of the **ip route-map** command to delete an entire route map, a route map sequence, or a specific statement within a sequence.

To delete an entire route map, enter **no ip route-map** followed by the route map name. For example, the following command deletes the entire route map named redistipv4:

```
-> no ip route-map redistipv4
```

To delete a specific sequence number within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the actual number. For example, the following command deletes sequence 10 from the redistipv4 route map:

```
-> no ip route-map redistipv4 sequence-number 10
```

Note that in the above example, the redistripv4 route map is not deleted. Only those statements associated with sequence 10 are removed from the route map.

To delete a specific statement within a route map, enter **no ip route-map** followed by the route map name, then **sequence-number** followed by the sequence number for the statement, then either **match** or **set** and the match or set parameter and value. For example, the following command deletes only the match tag 8 statement from route map redistipv4 sequence 10:

```
-> no ip route-map redistipv4 sequence-number 10 match tag 8
```

Setting the Metric

A route map can be used to set the metric by adding, subtracting, or replacing the metric of a route as in the example below:

```
-> ip route-map set-metric set metric 1 effect add
```

- Add - Adds the given value to the routes metric
- Subtract - Subtracts the given value from the metric (can't be less than 1)
- Replace - Uses the given value for the routes metric
- None - Ignores the given value and passes the routes metric through

Denying A Route

With route maps denying a route does not mean that all the other routes are automatically permitted. It is necessary to configure proper permit/deny rule for each route. However, a permit rule can be created to allow all routes and then specific rules for denying certain routes can be created as in the example below:

```
-> ip route-map leakin match ip-address 0.0.0.0/0 permit (permits all routes)
-> ip route-map leakin-example match ip-address 14.14.0.0/16 (deny route)
```

Configuring Route Map Sequences

A route map may consist of one or more sequences of statements. The sequence number determines which statements belong to which sequence and the order in which sequences for the same route map are processed.

To add match and set statements to an existing route map sequence, specify the same route map name and sequence number for each statement. For example, the following series of commands creates route map rm_1 and configures match and set statements for the rm_1 sequence 10:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 set metric 1
```

To configure a new sequence of statements for an existing route map, specify the same route map name but use a different sequence number. For example, the following command creates a new sequence 20 for the rm_1 route map:

```
-> ip route-map rm_1 sequence-number 20 action permit
-> ip route-map rm_1 sequence-number 20 match ipv4-interface to-finance
-> ip route-map rm_1 sequence-number 20 set metric 5
```

The resulting route map appears as follows:

```
-> show ip route-map rm_1
Route Map: rm_1 Sequence Number: 10 Action permit
  match tag 8
  set metric 1
Route Map: rm_1 Sequence Number: 20 Action permit
  match ip4 interface to-finance
  set metric 5
```

Sequence 10 and sequence 20 are both linked to route map rm_1 and are processed in ascending order according to their sequence number value. Note that there is an implied logical OR between sequences. As a result, if there is no match for the tag value in sequence 10, then the match interface statement in sequence 20 is processed. However, if a route matches the tag 8 value, then sequence 20 is not used. The set statement for whichever sequence was matched is applied.

A route map sequence may contain multiple match statements. If these statements are of the same kind (e.g., match tag 5, match tag 8, etc.) then a logical OR is implied between each like statement. If the match statements specify different types of matches (e.g., match tag 5, match ip4 interface to-finance, etc.), then a logical AND is implied between each statement. For example, the following route map sequence will redistribute a route if its tag is either 8 or 5:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
```

The following route map sequence will redistribute a route if the route has a tag of 8 or 5 *and* the route was learned on the IPv4 interface to-finance:

```
-> ip route-map rm_1 sequence-number 10 action permit
-> ip route-map rm_1 sequence-number 10 match tag 5
-> ip route-map rm_1 sequence-number 10 match tag 8
-> ip route-map rm_1 sequence-number 10 match ipv4-interface to-finance
```

Configuring Access Lists

An IP access list provides a convenient way to add multiple IPv4 or IPv6 addresses to a route map. Using an access list avoids having to enter a separate route map statement for each individual IP address. Instead, a single statement is used that specifies the access list name. The route map is then applied to all the addresses contained within the access list.

Configuring an IP access list involves two steps: creating the access list and adding IP addresses to the list. To create an IP access list, use the **ip access-list** command (IPv4) or the **ipv6 access-list** command (IPv6) and specify a name to associate with the list. For example:

```
-> ip access-list ipaddr
-> ipv6 access-list ip6addr
```

To add addresses to an access list, use the **ip access-list address** (IPv4) or the **ipv6 access-list address** (IPv6) command. For example, the following commands add addresses to an existing access list:

```
-> ip access-list ipaddr address 16.24.2.1/16
-> ipv6 access-list ip6addr address 2001::1/64
```

Use the same access list name each time the above commands are used to add additional addresses to the same access list. In addition, both commands provide the ability to configure if an address and/or its matching subnet routes are permitted or denied redistribution. For example:

```
-> ip access-list ipaddr address 16.24.2.1/16 action deny redist-control all-
subnets
-> ipv6 access-list ip6addr address 2001::1/64 action permit redist-control no-
subnets
```

For more information about configuring access list commands, see the "IP Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Route Map Redistribution

The **ip redist** command is used to configure the redistribution of routes from a source protocol into the BGP destination protocol. This command is used on the BGP router that will perform the redistribution.

A source protocol is a protocol from which the routes are learned. A destination protocol is the one into which the routes are redistributed. Make sure that both protocols are loaded and enabled before configuring redistribution.

Redistribution applies criteria specified in a route map to routes received from the source protocol. Therefore, configuring redistribution requires an existing route map. For example, the following command configures the redistribution of OSPF routes into the BGP network using the ospf-to-bgp route map:

```
-> ip redist ospf into bgp route-map ospf-to-bgp
```

OSPF routes received by the router interface are processed based on the contents of the ospf-to-bgp route map. Routes that match criteria specified in this route map are either allowed or denied redistribution into the BGP network. The route map may also specify the modification of route information before the route is redistributed. See "Using Route Maps" on page 4-44 for more information.

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

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

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

```
-> show ip redist

Source       Destination
Protocol     Protocol     Status    Route Map
-------------+------------+---------+--------------------
LOCAL4       RIP          Enabled   rip_1
LOCAL4       OSPF         Enabled   ospf_2
LOCAL4       BGP          Enabled   bgp_3
RIP          OSPF         Enabled   ospf-to-bgp
```

Configuring the Administrative Status of the Route Map Redistribution

To change the default administrative status of a route map redistribution configuration, use the **status** parameter with the **ip redist** command. For example, the following command disables the redistribution administrative status for the specified route map:

```
-> ip redist ospf into bgp route-map ospf-to-bgp admin-state disable
```

The following command example enables the administrative status:

```
-> ip redist ospf into rip route-map ospf-to-bgp admin-state enable
```

Route Map Redistribution Example

The following example configures the redistribution of OSPF routes into a BGP network using a route map (ospf-to-bgp) to filter specific routes:

```
-> ip route-map ospf-to-bgp sequence-number 10 action deny
-> ip route-map ospf-to-bgp sequence-number 10 match tag 5
-> ip route-map ospf-to-bgp sequence-number 10 match route-type external type2
-> ip route-map ospf-to-bgp sequence-number 20 action permit
-> ip route-map ospf-to-bgp sequence-number 20 match ipv4-interface intf_ospf
-> ip route-map ospf-to-bgp sequence-number 20 set metric 255

-> ip route-map ospf-to-bgp sequence-number 30 action permit
-> ip route-map ospf-to-bgp sequence-number 30 set tag 8

-> ip redist ospf into bgp route-map ospf-to-bgp
```

The resulting ospf-to-bgp route map redistribution configuration does the following:

- Denies the redistribution of Type 2 external BGP routes with a tag set to five.
- Redistributes into BGP all routes learned on the intf_ospf interface and sets the metric for such routes to 255.
- Redistributes all other routes (those not processed by sequence 10 or 20) and sets the tag for such routes to eight.

Configuring Redundant CMMs for Graceful Restart

On an OmniSwitch router in a redundant CMM configuration, inter-domain routing is not disrupted during a CMM takeover/failover. BGP retains routing information using Graceful Restart mechanisms and also helps a peering BGP router perform a BGP graceful restart. This supports the continuous forwarding of inter-domain traffic flows.

To configure BGP graceful restart support on OmniSwitch switches, use the **ip bgp graceful-restart** command by entering **ip bgp graceful-restart**.

For example, to support BGP graceful restart, enter:

```
-> ip bgp graceful-restart
```

To configure the grace period to achieve a graceful BGP restart, use the **ip bgp graceful-restart restart-interval** command, followed by the value in seconds.

For example, to configure a BGP graceful restart grace period as 300 seconds, enter:

```
-> ip bgp graceful-restart restart-interval 60
```

To disable support for graceful restart, use the **no** form of the **ip bgp graceful-restart** command by entering:

```
-> no ip bgp graceful-restart
```

For more information about graceful restart commands, see the “BGP Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Application Example

The following simple network using EBGP and IBGP will demonstrate some of the basic BGP setup commands discussed previously:

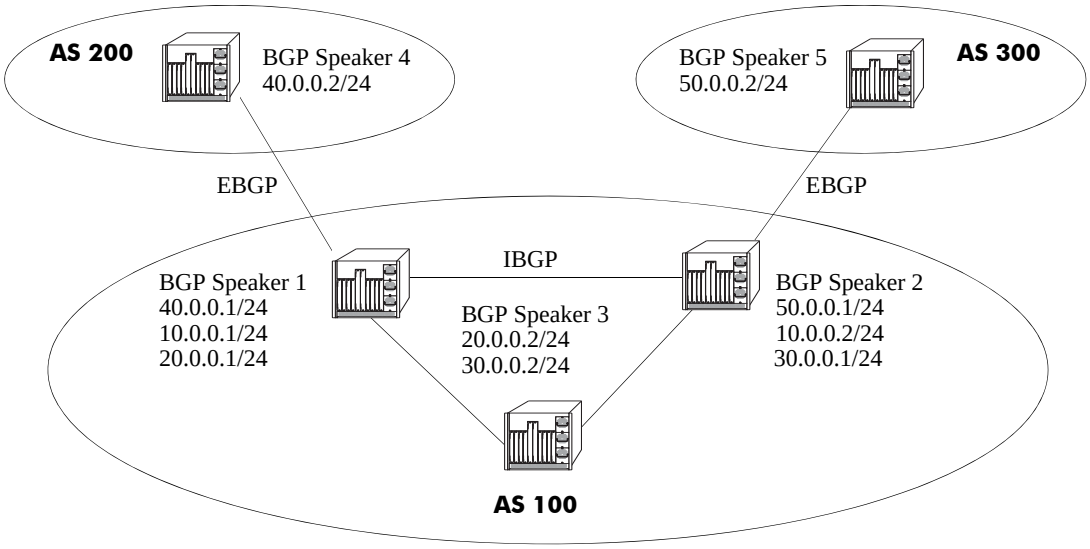


Figure 4-14 : BGP Application Example

In the above network, Speakers 1, 2, and 3 are part of AS 100 and are fully meshed. Speaker 4 is in AS 200 and Speaker 5 is in AS 300.

AS 100

BGP Speaker 1

Assign the speaker to AS 100:

```
-> ip bgp autonomous-system 100
```

Peer with the other speakers in AS 100 (for internal BGP, and to create a fully meshed BGP network):

```
-> ip bgp neighbor 20.0.0.2
-> ip bgp neighbor 20.0.0.2 remote-as 100
-> ip bgp neighbor 20.0.0.2 admin-state enable

-> ip bgp neighbor 10.0.0.2
-> ip bgp neighbor 10.0.0.2 remote-as 100
-> ip bgp neighbor 10.0.0.2 admin-state enable
```

Peer with the external speaker in AS 200 (for external BGP):

```
-> ip bgp neighbor 40.0.0.2
-> ip bgp neighbor 40.0.0.2 remote-as 200
-> ip bgp neighbor 40.0.0.2 admin-state enable
```

Administratively enable BGP:

```
-> ip bgp admin-state enable
```

BGP Speaker 2

Assign the speaker to AS 100:

```
-> ip bgp autonomous-system 100
```

Peer with the other speakers in AS 100 (for internal BGP, and to create a fully meshed BGP network):

```
-> ip bgp neighbor 30.0.0.2
-> ip bgp neighbor 30.0.0.2 remote-as 100
-> ip bgp neighbor 30.0.0.2 admin-state enable

-> ip bgp neighbor 10.0.0.1
-> ip bgp neighbor 10.0.0.1 remote-as 100
-> ip bgp neighbor 10.0.0.1 admin-state enable
```

Peer with the external speaker in AS 300 (for external BGP):

```
-> ip bgp neighbor 50.0.0.2
-> ip bgp neighbor 50.0.0.2 remote-as 300
-> ip bgp neighbor 50.0.0.2 admin-state enable
```

Administratively enable BGP:

```
-> ip bgp admin-state enable
```

BGP Speaker 3

Assign the speaker to AS 100:

```
-> ip bgp autonomous-system 100
```

Peer with the other speakers in AS 100 (for internal BGP, and to create a fully meshed BGP network):

```
-> ip bgp neighbor 30.0.0.1
-> ip bgp neighbor 30.0.0.1 remote-as 100
-> ip bgp neighbor 30.0.0.1 admin-state enable

-> ip bgp neighbor 20.0.0.1
-> ip bgp neighbor 20.0.0.1 remote-as 100
-> ip bgp neighbor 20.0.0.1 admin-state enable
```

Administratively enable BGP:

```
-> ip bgp admin-state enable
```

AS 200

BGP Speaker 4

Assign the speaker to AS 200:

```
-> ip bgp as 200
```

Peer with the external speaker in AS 100 (for external BGP):

```
-> ip bgp neighbor 40.0.0.1
-> ip bgp neighbor 40.0.0.1 remote-as 100
-> ip bgp neighbor 40.0.0.1 admin-state enable
```

Administratively enable BGP:

```
-> ip bgp admin-state enable
```

AS 300

BGP Speaker 5

Assign the speaker to AS 300:

```
-> ip bgp autonomous-system 300
```

Peer with the external speaker in AS 100 (for external BGP):

```
-> ip bgp neighbor 50.0.0.1
-> ip bgp neighbor 50.0.0.1 remote-as 100
-> ip bgp neighbor 50.0.0.1 admin-state enable
```

Administratively enable BGP:

```
-> ip bgp admin-state enable
```

Displaying BGP Settings and Statistics

Use the show commands listed in the following table to display information about the current BGP configuration and on BGP statistics:

show ip bgp	Displays the current global settings for the local BGP speaker.
show ip bgp statistics	Displays BGP global statistics, such as the route paths.
show ip bgp aggregate-address	Displays aggregate configuration information.
show ip bgp dampening	Displays the current route dampening configuration settings.
show ip bgp dampening-stats	Displays route flapping statistics.
show ip bgp network	Displays information on the currently defined BGP networks.
show ip bgp path	Displays information, such as Next Hop and other BGP attributes, for every path in the BGP routing table.
show ip bgp neighbors	Displays characteristics for BGP peers.
show ip bgp neighbors policy	Displays current inbound and outbound policies for all peers in the router.
show ip bgp neighbors timer	Displays current and configured values for BGP timers, such as the hold time, route advertisement, and connection retry.
show ip bgp neighbors statistics	Displays statistics, such as number of messages sent and received, for the peer.
show ip bgp policy aspath-list	Displays information on policies based on AS path criteria.
show ip bgp policy community-list	Displays information on policies based on community list criteria.
show ip bgp policy prefix-list	Displays information on policies based on route prefix criteria.
show ip bgp policy route-map	Displays information on currently configured route maps.
show ip redist	Displays the route map redistribution configuration.
show ip bgp routes	Displays information on BGP routes known to the router. This information includes whether changes to the route are in progress, whether it is part of an aggregate route, and whether it is dampened.

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

BGP for IPv6 Overview

IP version 6 (IPv6) is a new version of the Internet Protocol, designed as the successor to IP version 4 (IPv4), to overcome certain limitations in IPv4. IPv6 adds significant extra features that were not possible with IPv4. These include automatic configuration of hosts, extensive multicasting capabilities, and built-in security using authentication headers and encryption. Built-in support for QOS and path control are also features found in IPv6.

IPv6 is a hierarchical 128-bit addressing scheme that consists of 8 fields, comprising 16 bits each. An IPv6 address is written as a hexadecimal value (0-F) in groups of four, separated by colons. IPv6 provides 3x10^38 addresses, which can help overcome the shortage of IP addresses needed for internet usage.

There are three types of IPv6 addresses: Unicast, Anycast, and Multicast. A Unicast address identifies a single interface and a packet destined for a Unicast address is delivered to the interface identified by that address. An Anycast address identifies a set of interfaces and a packet destined for an Anycast address is delivered to the nearest interface identified by that Anycast address. A Multicast address identifies a set of interfaces and a packet destined for a Multicast address is delivered to all the interfaces identified by that Multicast address. There are no broadcast addresses in IPv6.

BGP uses Multiprotocol Extensions to support IPv6. The same procedures used for IPv4 prefixes can be applied for IPv6 prefixes as well and the exchange of IPv4 prefixes will not be affected by this new feature. However, there are some attributes that are specific to IPv4, such as AGGREGATOR, NEXT_HOP and NLRI. Multiprotocol Extensions for BGP also supports backward compatibility for the routers that do not support this feature.

To enable this implementation of BGP to support routing for multiple Network Layer protocols (e.g., IPv6, etc.), the following capabilities are added:

- Associating a particular Network Layer protocol with the next hop information.
- Associating a particular Network Layer protocol with NLRI.

To support Multiprotocol BGP Extensions, two new non-transitive attributes are introduced, Multiprotocol Reachable NLRI (MP_REACH_NLRI) and Multiprotocol Unreachable NLRI (MP_UNREACH_NLRI). MP_REACH_NLRI is utilized to carry the set of reachable destinations along with the next hop information to be used for these destinations. The MP_UNREACH_NLRI attribute carries the set of unreachable destinations.

Multiprotocol BGP extensions support the advertisement of IPv6 prefixes over the BGP sessions established between two BGP speakers using either of their IPv4 or IPv6 addresses. IPv6 prefixes can be redistributed into BGP using route maps. Similar to IPv4 networks, IPv6 networks should also be injected into BGP for a BGP speaker to advertise the network to its peers.

Some features that are not supported in the current release of Multiprotocol BGP include:

- IPv6 route-flap dampening
- IPv6 route aggregation
- Other multiprotocol capabilities for VPNs, MPLS label exchanges, and so on.

Quick Steps for Using BGP for IPv6

The following steps create an IPv4 BGP peer capable of exchanging IPv6 prefixes:

1 The BGP software is not loaded automatically when the router is booted. You must manually load the software into memory by typing the following command:

```
-> ip load bgp
```

2 Assign an Autonomous System (AS) number to the local BGP speaker in this router. You can change the default AS number to fit your network requirements. For example:

```
-> ip bgp autonomous-system 100
```

3 To enable unicast IPv6 updates for the BGP routing process, use the following command:

```
-> ipv6 bgp unicast
```

4 Create an IPv4 BGP peer entry. The local BGP speaker should be able to reach this peer. The IPv4 address you assign the peer should be valid. For example:

```
-> ip bgp neighbor 23.23.23.23
```

5 Assign an AS number to the IPv4 BGP peer you just created. All peers require an AS number. The AS number does not have to be the same as the AS number for the local BGP speaker. For example:

```
-> ip bgp neighbor 23.23.23.23 remote-as 200
```

6 To enable the exchange of IPv6 unicast prefixes between IPv4 BGP peers, use the following command:

```
-> ip bgp neighbor 23.23.23.23 activate-ipv6
```

7 Configure the IPv6 next hop address for the IPv6 prefixes advertised to the IPv4 BGP peer using the following command:

```
-> ip bgp neighbor 23.23.23.23 ipv6-nexthop 2001:100:3:4::1
```

Note. *Optional*. To reset the IPv6 next hop value, use an all-zero address. For example:

```
 -> ip bgp neighbor 23.23.23.23 ipv6-nexthop::
```

For more information, refer to the *OmniSwitch AOS Release 8 CLI Reference Guide*.

---

8 As an IPv4 BGP peer is not active on the network until you enable it, use the following command to enable the IPv4 peer created in Step 4:

```
-> ip bgp neighbor 23.23.23.23 admin-state enable
```

9 Administratively enable BGP using the following command:

```
-> ip bgp admin-state enable
```

The following steps create an IPv6 BGP peer capable of exchanging IPv6 prefixes:

1 Repeat steps 1 through 3 from the previous section to load the BGP software, assign an AS number to the local BGP speaker, and enable unicast IPv6 updates for the BGP routing process, respectively.

2 Create an IPv6 BGP peer entry. The local BGP speaker should be able to reach this peer. The IPv6 address you assign the peer should be valid. For example:

```
-> ipv6 bgp neighbor 2001:100:3:4::1
```

3 Assign an AS number to the IPv6 BGP peer you just created. All peers require an AS number. The AS number does not have to be the same as the AS number for the local BGP speaker. For example:

```
-> ipv6 bgp neighbor 2001:100:3:4::1 remote-as 10
```

4 To enable the exchange of IPv6 unicast prefixes between IPv6 BGP peers, use the following command:

```
-> ipv6 bgp neighbor 2001:100:3:4::1 activate-ipv6
```

5 As an IPv6 BGP peer is not active on the network until you enable it, use the following command to enable the IPv6 peer created in Step 2:

```
-> ipv6 bgp neighbor 2001:100:3:4::1 admin-state enable
```

6 Administratively enable BGP using the following command:

```
-> ip bgp admin-state enable
```

Note. In homogeneous IPv6 networks (i.e., in the absence of IPv4 interface configuration), the router's router ID and the primary address must be explicitly configured prior to configuring the BGP protocol. This is because the router ID is a unique 32-bit identifier and the primary address is a unique IPv4 address that identifies the router. BGP uses the primary address in the AGGREGATOR attribute.

Configuring BGP for IPv6

This section describes the BGP for IPv6 configuration, which includes enabling and disabling IPv6 BGP unicast, configuring IPv6 BGP peers, and configuring IPv6 BGP networks using the OmniSwitch Command Line Interface (CLI) commands.

Enabling/Disabling IPv6 BGP Unicast

As BGP peers exchange only IPv4 unicast address prefixes, in order to exchange other address prefix types, such as IPv6 prefixes, you need to enable IPv6 unicast advertisements.

To enable IPv6 unicast updates, use the **ipv6 bgp unicast** command, as shown:

```
-> ipv6 bgp unicast
```

In a homogenous IPv6 network, you need to first disable the IPv4 unicast updates, and then enable the IPv6 unicast updates.

To disable IPv4 unicast updates, use the **no** form of the **ip bgp unicast** command, as shown:

```
-> no ip bgp unicast
```

Now, you can enable IPv6 unicast updates.

However, in IPv6 environments where the BGP speakers have established peering using their IPv4 addresses, IPv4 unicasting may not be disabled.

Configuring an IPv6 BGP Peer

A router configured to run the BGP routing protocol is called a BGP speaker. Unlike some other routing protocols, BGP speakers do not automatically discover each other and begin exchanging information. Instead, each BGP speaker must be explicitly configured with a set of BGP neighbors to exchange routing information. BGP is connection-oriented and uses TCP to establish a reliable connection. An underlying connection between two BGP speakers is established before any routing information is exchanged.

BGP supports two types of peers or neighbors, internal and external. Internal sessions run between BGP speakers in the same autonomous system. External sessions run between BGP peers in different autonomous systems.

Every BGP speaker should be assigned to an AS. A BGP speaker can be configured as a peer within the same or different AS.

You can configure BGP speakers to exchange IPv6 prefixes using either their IPv4 or IPv6 addresses. As BGP speakers exchange only IPv4 unicast address prefixes, in order to exchange other address prefix types, such as IPv6 prefixes, BGP speakers must be activated to advertise IPv6 BGP prefixes.

BGP peering can be established using either IPv4 or IPv6 addresses. However, in the absence of IPv4 interface configuration, it is mandatory to explicitly configure the router's router ID and assign a unique IPv4 address as the router's primary address.

Note. In this document, the BGP terms “peer” and “neighbor” are used interchangeably to mean any BGP entity known to the local router.

IPv6 Peer Command Defaults

The following table lists the default values for many of the peer commands:

Parameter Description	Command	Default Value/ Comments
Enables or disables BGP peer.	**ipv6 bgp neighbor admin-state**	disabled
Assigns an AS number to the BGP peer.	**ipv6 bgp neighbor remote-as**	1
Configures the time interval for updates between external BGP peers.	**ipv6 bgp neighbor advertisement-interval**	30 seconds
Enables or disables BGP peer automatic restart.	**ipv6 bgp neighbor auto-restart**	enabled
Configures this peer as a client to the local route reflector.	**ipv6 bgp neighbor route-reflector-client**	disabled
The interval, in seconds, between BGP retries to set up a connection via the transport protocol with another peer.	**ipv6 bgp neighbor conn-retry-interval**	120 seconds
Enables or disables BGP peer default origination.	**ipv6 bgp neighbor default-originate**	disabled
Configures the tolerated hold time interval, in seconds, for this peer's session, and timer interval between KEEPALIVE messages sent to this peer.	**ipv6 bgp neighbor timers**	hold time - 90 seconds keep alive - 30 seconds
Configures the maximum number of prefixes, or paths, the local router can receive from this peer in UPDATE messages.	**ipv6 bgp neighbor maximum-prefix**	5000
Enable or disables maximum prefix warning for a peer.	**ipv6 bgp neighbor maximum-prefix warning-only**	80 percent
Allows external peers to communicate with each other even when they are not directly connected.	**ipv6 bgp neighbor ebgp-multihop**	disabled
Configures the BGP peer name.	**ipv6 bgp neighbor description**	peer IPv6 address
Sets the BGP peer to use its own peering address as the next hop in UPDATE messages.	**ipv6 bgp neighbor next-hop-self**	disabled
Configures the local BGP speaker to wait for this peer to establish a connection.	**ipv6 bgp neighbor passive**	disabled

Parameter Description	Command	Default Value/ Comments
Enables or disables the stripping of private autonomous system numbers from the AS path of routes destined to this peer.	**ipv6 bgp neighbor remove-private-as**	disabled
Enables or disables BGP peer soft reconfiguration.	**ipv6 bgp neighbor soft-reconfiguration**	enabled
Configures this peer as a member of the same confederation as the local BGP speaker.	**ip bgp confederation neighbor**	disabled
Configures the local transport endpoint address for this neighbor's peering session.	**ipv6 bgp neighbor update-source**	Not set until configured
Configures the local IPv6 interface from which the peer will be connected if the peer is configured with its link-local address.	**ipv6 bgp neighbor update-source-address**	Not set until configured
Configures the IPv4 next hop addresses for IPv4 prefixes advertised between BGP peers.	**ipv6 bgp neighbor ipv4-nexthop**	IPv4 next hop value is set to all zeros.

BGP Peer Behavior using Local IPv6 Unicast Addresses

- The local IPv6 address prefixes are exchanged between internal BGP (IBGP) speakers within the same Autonomous System (AS), unless denied by explicit policy configuration.
- As Exterior BGP (EBGP) peers between different AS ignore receipt of and do not advertise prefixes with the well-known FC00::/7 prefix, prefixes longer than FC00::/7 can be configured for inter-site communication.
- There may be specific /48 or longer routes created for one or more Local IPv6 prefixes. In such a case, explicit BGP configuration of peer policies must be configured to control learning/advertising of such prefixes.

Configuring an IPv4 BGP Peer to Exchange IPv6 Prefixes

A BGP peer that is identified by its IPv4 address can be used to exchange IPv6 prefixes. However, to do this both the peers should be enabled with IPv6 BGP unicast and should have interfaces that support IPv6 addresses. To configure an IPv4 BGP peer to exchange IPv6 prefixes, follow the steps mentioned below:

1 Create an IPv4 BGP peer with which the BGP speaker will establish peering using its IPv4 address with the **ip bgp neighbor** command, as shown:

```
-> ip bgp neighbor 190.17.20.16
```

2 Assign an AS number to the IPv4 peer using the **ip bgp neighbor remote-as** command. For example, to assign the peer created in Step 1 to AS number 200, you would enter:

```
-> ip bgp neighbor 190.17.20.16 remote-as 200
```

3 Enable IPv6 unicast capability for the IPv4 BGP peer using the **ip bgp neighbor activate-ipv6** command, as shown:

```
-> ip bgp neighbor 190.17.20.16 activate-ipv6
```

4 Set the IPv6 next hop address for IPv6 prefixes advertised to the IPv4 BGP peer using the **ip bgp neighbor ipv6-nexthop** command, as shown:

```
-> ip bgp neighbor 190.17.20.16 ipv6-nexthop 2001::1
```

5 Enable the BGP peer status using the **ip bgp neighbor admin-state** command. For example, to enable the status of the IPv4 BGP peer with an IPv4 address of 190.17.20.16, you would enter:

```
-> ip bgp neighbor 190.17.20.16 admin-state enable
```

Configuring an IPv6 BGP Peer to Exchange IPv6 Prefixes

To configure an IPv6 BGP peer to exchange IPv6 prefixes, follow the steps mentioned below:

1 Create an IPv6 BGP peer with which the BGP speaker will establish peering using its IPv6 address with the **ipv6 bgp neighbor** command, as shown:

```
-> ipv6 bgp neighbor 2001::1
```

2 Assign an AS number to the IPv6 peer using the **ipv6 bgp neighbor remote-as** command. For example, to assign the peer created in Step 1 to AS number 10, you would enter:

```
-> ipv6 bgp neighbor 2001::1 remote-as 10
```

3 Enable IPv6 unicast capability for the IPv6 BGP peer using the **ipv6 bgp neighbor activate-ipv6** command, as shown:

```
-> ipv6 bgp neighbor 2001::1 activate-ipv6
```

4 Enable the BGP peer status using the **ipv6 bgp neighbor admin-state** command. For example, to enable the status of the IPv6 BGP peer with an IPv6 address of `2001::1`, you would enter:

```
-> ipv6 bgp neighbor 2001::1 admin-state enable
```

Configuring an IPv6 BGP Peer Using Link-Local IPv6 Addresses to Exchange IPv6 Prefixes

To configure an IPv6 BGP peer using its link-local IPv6 address to exchange IPv6 prefixes, follow the steps mentioned below:

1 Create an IPv6 BGP peer with which the BGP speaker will establish peering using its link-local IPv6 address with the **ipv6 bgp neighbor** command, as shown:

```
-> ipv6 bgp neighbor fe80::2d0:95ff:fee2:6ed0
```

2 Assign an AS number to the IPv6 peer using the **ipv6 bgp neighbor remote-as** command. For example, to assign the peer created in Step 1 to AS number 20, you would enter:

```
-> ipv6 bgp neighbor fe80::2d0:95ff:fee2:6ed0 remote-as 20
```

3 Configure the local IPv6 interface from which the BGP peer will be reachable using the **ipv6 bgp neighbor update-source** command. For example, to configure Vlan2 as the IPv6 interface name from which the BGP peer is connected, you would enter:

```
-> ipv6 bgp neighbor fe80::2d0:95ff:fee2:6ed0 update-source Vlan2
```

4 Enable IPv6 unicast capability to the IPv6 BGP peer using the **ip bgp neighbor activate-ipv6** command, as shown:

```
-> ipv6 bgp neighbor fe80::2d0:95ff:fee2:6ed0 activate-ipv6
```

5 Enable the BGP peer status using the **ipv6 bgp neighbor admin-state** command. For example, to enable the status of the BGP peer with a link-local IPv6 address of fe80::2d0:95ff:fee2:6ed0, you would enter,

```
-> ipv6 bgp neighbor fe80::2d0:95ff:fee2:6ed0 admin-state enable
```

Configuring an IPv6 BGP Peer to Exchange Globally Unique IPv6 Unicast Addresses

By default, globally unique IPv6 unicast addresses are exchanged between internal BGP IPv6 peers. Exchange of globally unique IPv6 unicast addresses between external BGP IPv6 peers must be explicitly configured using BGP policy on both the BGP speakers.

To configure an IPv6 BGP Unique IPv6 Unicast Addresses follow the steps mentioned below:

1 Create a prefix list for the well-known Unique IPv6 Unicast address using the **ip bgp policy prefix6-list** as shown:

```
-> ip bgp policy prefix6-list uniqLocal FC00::/48
-> ip bgp policy prefix6-list uniqLocal FC00::/48 action permit
-> ip bgp policy prefix6-list uniqLocal FC00::/48 admin-state enable
```

2 Create an IPv6 BGP peer with which the BGP speaker will establish peering using the **ipv6 bgp neighbor** command, as shown:

```
-> ipv6 bgp neighbor 2021::10
```

3 Assign an AS number to the IPv6 peer using the **ipv6 bgp neighbor remote-as** command. For example, to assign the peer created in Step 2 to AS number 20, you would enter:

```
-> ipv6 bgp neighbor 2021::10 remote-as 20
```

4 Enable IPv6 unicast capability to the IPv6 BGP peer using the **ipv6 bgp neighbor** command, as shown:

```
-> ipv6 bgp neighbor 2021::10 activate-ipv6
```

5 Apply the policy to the bgp neighbor using the **ipv6 bgp neighbor in-prefix6list** and **ipv6 bgp neighbor out-prefix6list** commands as shown:

```
-> ipv6 bgp neighbor 2021::10 out-prefix6list uniqLocal
-> ipv6 bgp neighbor 2021::10 in-prefix6list uniqLocal
```

6 Enable the BGP peer status using the **ipv6 bgp neighbor admin-state** command:

```
-> ipv6 bgp neighbor 2021::10 admin-state enable
```

Configuring an IPv6 BGP peer to Exchange IPv4 Prefixes

A BGP peer that is identified by its IPv6 address can be used to exchange IPv4 prefixes. However, to do this, both peers should be enabled with IPv4 BGP unicast and should have interfaces that support IPv4 addresses. To configure an IPv6 BGP peer to exchange IPv4 prefixes, follow the steps mentioned below:

1 Create an IPv6 BGP peer with which the BGP speaker will establish peering using its IPv6 address with the **ipv6 bgp neighbor** command, as shown:

```
-> ipv6 bgp neighbor 2001::1
```

2 Assign an AS number to the IPv6 peer using the **ipv6 bgp neighbor remote-as** command. For example, to assign the peer created in Step 1 to AS number 10, you would enter:

```
-> ipv6 bgp neighbor 2001::1 remote-as 10
```

3 Set the IPv4 next hop address for IPv4 prefixes advertised to the IPv6 BGP peer using the **ipv6 bgp neighbor ipv4-nexthop** command, as shown:

```
-> ipv6 bgp neighbor 2001::1 ipv4-nexthop 190.17.20.1
```

4 Enable the BGP peer status using the **ipv6 bgp neighbor admin-state** command, as shown:

```
-> ipv6 bgp neighbor 2001::1 admin-state enable
```

Optional IPv6 BGP Peer Parameters

Peer Parameter	Command
Enables or disables BGP peer.	**ipv6 bgp neighbor admin-state**
Assigns an AS number to the BGP peer.	**ipv6 bgp neighbor remote-as**
Configures the time interval for updates between external BGP peers.	**ipv6 bgp neighbor advertisement-interval**
Enables or disables BGP peer automatic restart.	**ipv6 bgp neighbor auto-restart**
The interval, in seconds, between BGP retries to set up a connection via the transport protocol with another peer.	**ipv6 bgp neighbor conn-retry-interval**
Enables or disables BGP peer default origination.	**ipv6 bgp neighbor default-originate**
Configures the tolerated hold time interval, in seconds, for this peer's session, and timer interval between KEEPALIVE messages sent to this peer.	**ipv6 bgp neighbor timers**
Configures the maximum number of prefixes, or paths, the local router can receive from this peer in UPDATE messages.	**ipv6 bgp neighbor maximum-prefix**
Enable or disables maximum prefix warning for a peer.	**ipv6 bgp neighbor maximum-prefix warning-only**
Allows external peers to communicate with each other even when they are not directly connected.	**ipv6 bgp neighbor ebgp-multihop**
Sets the BGP peer to use its own peering address as the next hop in UPDATE messages.	**ipv6 bgp neighbor next-hop-self**
Configures the local BGP speaker to wait for this peer to establish a connection.	**ipv6 bgp neighbor passive**
Enables or disables the stripping of private autonomous system numbers from the AS path of routes destined to this peer.	**ipv6 bgp neighbor remove-private-as**
Enables or disables BGP peer soft reconfiguration.	**ipv6 bgp neighbor soft-reconfiguration**
Configures this peer as a member of the same confederation as the local BGP speaker.	**ip bgp confederation neighbor**

Peer Parameter	Command
Configures the local transport endpoint address for this neighbor's peering session.	**ipv6 bgp neighbor update-source**
Configures the local IPv6 interface from which the peer will be connected if the peer is configured with its link-local address.	**ipv6 bgp neighbor update-source-address**
Configures the IPv4 next hop addresses for IPv4 prefixes advertised between BGP peers.	**ipv6 bgp neighbor ipv4-nexthop**
Configures the check for the first AS in the ASPATH list while processing UPDATE message from BGP neighbor	**ipv6 bgp neighbor check-first-as**

Restarting a Peer

Many BGP peer commands will automatically restart the peer once they are configured. By restarting the peer, these parameters take effect as soon as the peer comes back up. However, there are some peer commands (such as those configuring timer values) that do not reset the peer. If you want these parameters to take effect, then you must manually restart the BGP peer using the **ipv6 bgp neighbor clear** command. The following command would restart the peer at address 190.17.20.16:

```
-> ipv6 bgp neighbor 2001::1 clear
```

The peer is not available to send or receive update or notification messages while it is restarting.

Use the **ipv6 bgp neighbor clear soft** command to reset all inbound or outbound policies to existing routes without restarting the peer session.

Setting the Peer Auto Restart

When the auto restart is enabled, this IPv6 peer will automatically attempt to restart a session with another peer after a session with that peer terminates.

To enable the auto restart feature, enter the **ipv6 bgp neighbor auto-restart** command with the peer IP address, as shown:

```
-> ipv6 bgp neighbor 2001::1 auto-restart
```

To disable this feature, enter the following:

```
-> no ipv6 bgp neighbor 2001::1 auto-restart
```

Changing the Local Router Address for an IPv6 Peer Session

TCP connections to an IPv6 peer's address are assigned to the closest interface based on reachability. Any operational local IPv6 address can be assigned to the IPv6 BGP peering session by explicitly forcing the TCP connection to use the specified address.

The **ipv6 bgp neighbor update-source-address** command sets the local IPv6 interface address through which this BGP peer can be contacted.

For example, to configure a peer which has an IPv6 address 2001::1 to be contacted through the local IPv6 address 2004::2, use the **ipv6 bgp neighbor update-source-address** command, as shown:

```
-> ipv6 bgp neighbor 2001::1 update-source-address 2004::2
```

Use the no form of the **ipv6 bgp neighbor update-source-address** command to prevent the peer with an IPv6 address of 2001::1 from contacting the speaker via the IPv6 address 2004::2, as shown:

```
-> no ipv6 bgp neighbor 2001::1 update-source-address
```

Note. If a BGP peer is configured with its link-local address, use the 'update-source' parameter to specify the name of the IPv6 interface from which this peer is reachable. For example, to configure a peer with a link-local address of fe80::2d0:95ff:fee2:6ed0 to be contacted through the IPv6 interface ipv6IntfVlan2, use the **ipv6 bgp neighbor update-source** command, as shown:

```
-> ipv6 bgp neighbor fe80::2d0:95ff:fee2:6ed0 update-source ipv6IntfVlan2
```

Clearing Statistics for a Peer

BGP tracks the number of messages sent to and received from other peers. It also breaks down messages into UPDATE, NOTIFICATION, and TRANSITION categories. You can reset, or clear, the statistics for a peer using the **ipv6 bgp neighbor stats-clear** command. For example, use of the **ipv6 bgp neighbor stats-clear** command would clear statistics for the IPv6 peer at address 2001::2.

```
-> ipv6 bgp neighbor 2001::2 stats-clear
```

The statistics that are cleared are shown in the **show ipv6 bgp neighbors statistics** command. The following is an example of output from this command:

```
-> show ipv6 bgp neighbors statistics
Legends: Nbr    = Neighbor
         As     = Autonomous System
         RMSGS = # of received messages
         SMSGS = # of sent messages
         RUPDS = # of Update messages received
         SUPDS = # of Update messages sent
         RNOFY = # of Notify messages received
         SNOFY = # of Notify messages sent
         RPFXS = # of prefixes received
         UPTNS = # of UP transitions
         DNTNS = # of DOWN transitions

Nbr address              As    RMSGS SMSGS RUPDS SUPDS RNOFY SNOFY RPFXS UPTNS DNTNS
-----------------------+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
2001:100:3:4::1          30    225   260   2     3     0     0     10    1     1
```

Setting the Peer Route Advertisement Interval

The route advertisement interval specifies the frequency at which routes external to the autonomous system are advertised. These advertisements are also referred to as UPDATE messages. This interval applies to advertisements to external IPv6 BGP peers.

To set the advertisement interval, enter the number of seconds in conjunction with the **ipv6 bgp neighbor advertisement-interval** command as shown:

```
-> ipv6 bgp neighbor 2001::1 advertisement-interval 60
```

The interval is now set to 60 seconds.

Configuring a BGP Peer with the IPv6 Loopback0 Interface

An IPv6 Loopback0 virtual interface is used to identify a consistent IPv6 address for network management purposes. The IPv6 Loopback0 interface is not bound to any VLAN or other physical interface, so it will always remain operationally active. This differs from other IPv6 interfaces, If there are no active ports in the VLAN, all IPv6 interfaces associated with that VLAN are not active. In addition, the IPv6 Loopback0 interface provides a unique IPv6 address for the switch that is easily identifiable to network management applications.

It is possible to create BGP peers using the IPv6 Loopback0 interface address of the peering router and binding the source address (that is, outgoing IPv6 address for the TCP connection) to its own configured IPv6 Loopback0 interface address. The IPv6 Loopback0 interface address can be used for both internal and external BGP peer sessions. For EBGP sessions, if the external peer router is multiple hops away, the **ebgp-multihop** parameter may need to be used.

The following example configures a BGP peering session using an IPv6 Loopback0 interface which is configured with the address 2004::2:

```
-> ipv6 bgp neighbor 2001::1 update-source-address 2004::2
```

See the *OmniSwitch AOS Release 8 Network Configuration Guide* for more information about configuring an IPv6 Loopback0 interface.

Configuring the advertising of IPv4 routes for IPv6 peers

You can enable or disable the advertising of IPv4 routes to an IPv6 neighbor. To enable the advertisement of IPv4 unicast capability to the IPv6 BGP peer, use the following command.

```
-> ipv6 bgp neighbor 2001::1 activate-ipv4
```

The advertising capability is enabled for the IPv6 BGP peer with IPv6 address 2001::1.

Note. The advertisement of IPv4 unicast capability is enabled by default.

To disable the advertising capability use the no form of the command.

```
-> no ipv6 bgp neighbor 2001::1 activate-ipv4
```

Setting Peer Authentication

You can set which MD5 authentication key this router will use when contacting a peer. To set the MD5 authentication key, enter the peer IPv6 address and key with the **ipv6 bgp neighbor md5 key** command:

```
-> ipv6 bgp neighbor 2001::1 md5 key openpeer2
```

The peer with IPv6 address 2001::1 will be sent messages using “openpeer2” as the encryption password. If this is not the password set on peer 2001::1, then the local router will not be able to communicate with this peer.

Configuring IPv6 BGP Networks

A local IPv6 BGP network is used to indicate to BGP that a network should originate from a specified router. A network must be known to the local BGP speaker and must also originate from the local BGP speaker.

Networks have certain parameters that can be configured, such as **local-preference**, **community**, **metric**, etc. Note that the network specified must be known to the router, whether it is connected, static, or dynamically learned. This is not the case for an aggregate.

Adding a Network

To add a local network to a BGP speaker, use the IPv6 address and mask of the local network in conjunction with the **ipv6 bgp network** command, as shown:

```
-> ipv6 bgp network 2001::/64
```

In this example, the network 2001::/64 is the local IPv6 network for this BGP speaker.

To remove the same network from the BGP speaker, use the **no** form of the **ipv6 bgp network** command, as shown:

```
-> no ipv6 bgp network 2001::/64
```

The network will now no longer be associated as the local network for the BGP speaker.

Enabling a Network

Once the network has been added to the speaker, it must be enabled on the speaker. To do this, enter the IPv6 address and mask of the local network in conjunction with the **ipv6 bgp network admin-state** command, as shown:

```
-> ipv6 bgp network 2001::/64 admin-state enable
```

In this example, the IPv6 network 2001::/64 has now been enabled.

To disable the same network, enter the **ipv6 bgp network admin-state** command, as shown:

```
-> ipv6 bgp network 2001::/64 admin-state disable
```

The network would now be disabled, though not removed from the speaker.

Configuring Network Parameters

Once a local IPv6 network is added to a speaker, you can configure three parameters that are attached to routes generated by the **ipv6 bgp network** command. These three attributes are the local preference, community, and route metric.

Local Preference

Local preference is an attribute that specifies the degree of preference to be given to a specific route when there are multiple routes to the same destination. This attribute is propagated throughout the autonomous system and is represented by a numeric value. The higher the number, the higher the preference. For example, a route with two exits, one with a local preference of 50 and another with a local preference 30 will use the path which has the local preference of 50.

To change from the default local preference for the local network, enter the IPv6 address and mask of the local network in conjunction with the **ipv6 bgp network local-preference** command and value, as shown:

```
-> ipv6 bgp network 2001::/64 local-preference 600
```

The local preference for routes generated by the network is now changed from the default value to 600.

Community

Communities are a way of grouping BGP destination addresses that share some common property. Adding the local network to a specific community indicates that the network shares a common set of properties with the rest of the community.

To change from the default community and add a network to a community, enter the local network IPv6 address and mask in conjunction with the **ipv6 bgp network community** command and name, as shown:

```
-> ipv6 bgp network 2001::/64 community 100:200
```

Network 2001::/64 is now changed from the default community to the 100:200 community.

To remove the local network from the community, enter the local network as above with the community set to “none”, as shown:

```
-> ipv6 bgp network 2001::/64 community none
```

The network is now no longer in any community.

Metric

A metric for an IPv6 network is the Multi-Exit Discriminator (MED) value. This value is sent from routers of one AS to another to indicate the path that the remote AS can use to send data to the local AS assuming there is more than one. A lower value indicates a more preferred exit point. For example, a route with a MED of 10 is more likely to be used than a route with an MED of 100.

To change from the default network metric value and set a new network metric value, enter the network IPv6 address and mask in conjunction with the **ipv6 bgp network metric** command and value, as shown:

```
-> ipv6 bgp network 2001::/64 metric 100
```

The IPv6 network 2001::/64 is now changed from the default metric to the new metric of 100.

Viewing Network Settings

To view the network settings for all IPv6 networks assigned to the speaker, enter the **show ipv6 bgp network** command, as shown:

```
-> show ipv6 bgp network
```

A display similar to the following appears:

```
Network                Admin state Oper state
----------------------+-----------+------------
2525:500:600::/64       enabled    active
```

To display a specific IPv6 network, enter the same command with the network IPv6 address and mask, as shown:

```
-> show ipv6 bgp network 2525:500:600::/64.
```

A display similar to the following appears:

```
Network address          = 2525:500:600::/64,
Network admin state      = enabled,
Network oper state       = active,
Network metric           = 0,
Network local preference = 0,
Network community string = <none>
```

Configuring IPv6 Redistribution

It is possible to learn and advertise IPv6 routes between different routing protocols. Such a process is referred to as route redistribution and is configured using the **ipv6 redist** command.

IPv6 redistribution uses route maps to control how external routes are learned and distributed. A route map consists of one or more user-defined statements that can determine which routes are allowed or denied access to the network. In addition, a route map may also contain statements that modify route parameters before they are redistributed.

When a route map is created, it is given a name to identify the group of statements that it represents. This name is required by the **ipv6 redist** command. Therefore, configuring IPv6 BGP route redistribution involves the following steps:

1 Create a route map, as described in "Using Route Maps for IPv6 Redistribution" on page 4-71.

2 Configure IPv6 redistribution to apply a route map, as described in "Configuring IPv6 Route Map Redistribution" on page 4-71.

Using Route Maps for IPv6 Redistribution

A route map specifies the criteria that are used to control redistribution of routes between protocols. Route maps that are used for redistributing both IPv4 and IPv6 routes are created in the same way. Refer to "Using Route Maps" on page 4-44 for more information.

Configuring IPv6 Route Map Redistribution

Once a route map is created, it is then applied using the **ipv6 redist** command. The **ipv6 redist** command is used to configure the redistribution of routes from a source protocol into the IPv6 BGP destination protocol. This command is used on the IPv6 BGP router that will perform the redistribution.

A source protocol is a protocol from which the routes are learned. A destination protocol is the one into which the routes are redistributed. Make sure that both protocols are loaded and enabled before configuring redistribution.

Redistribution applies criteria specified in a route map to routes received from the source protocol. Therefore, configuring redistribution requires an existing route map. For example, the following command configures the redistribution of OSPFv3 routes into the IPv6 BGP network using the ospf-to-bgp route map:

```
-> ipv6 redist ospf into bgp route-map ospf-to-bgp
```

OSPFv3 routes received by the router interface are processed based on the contents of the ospf-to-bgp route map. Routes that match criteria specified in this route map are either allowed or denied redistribution into the IPv6 BGP network. The route map may also specify the modification of route information before the route is redistributed. See "Using Route Maps" on page 4-44 for more information.

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

```
-> no ipv6 redist ospf into bgp route-map ospf-to-bgp
```

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

```
-> show ipv6 redist

Source       Destination
Protocol     Protocol     Status    Route Map
-----------+------------+---------+--------------------
localIPv6    BGP          Enabled    ipv6rm
OSPFv3       RIPng        Enabled    ospf-to-rip
```

Configuring the Administrative Status of the Route Map Redistribution

To change the default administrative status of a route map redistribution configuration, use the **status** parameter with the **ipv6 redist** command. For example, the following command disables the redistribution administrative status for the specified route map:

```
-> ipv6 redist ospf into bgp route-map ospf-to-bgp admin-state disable
```

The following command example enables the administrative status:

```
-> ipv6 redist ospf into bgp route-map ospf-to-bgp admin-state enable
```

Route Map Redistribution Example

The following example configures the redistribution of OSPFv3 routes into an IPv6 BGP network using a route map (ospf-to-bgp) to filter specific routes:

```
-> ip route-map ospf-to-bgp sequence-number 10 action deny
-> ip route-map ospf-to-bgp sequence-number 10 match tag 5
-> ip route-map ospf-to-bgp sequence-number 10 match route-type external type2

-> ip route-map ospf-to-bgp sequence-number 20 action permit
-> ip route-map ospf-to-bgp sequence-number 20 match ipv6-interface intf_ospf
-> ip route-map ospf-to-bgp sequence-number 20 set metric 255

-> ip route-map ospf-to-bgp sequence-number 30 action permit
-> ip route-map ospf-to-bgp sequence-number 30 set tag 8

-> ipv6 redist ospf into bgp route-map ospf-to-bgp
```

The resulting ospf-to-bgp route map redistribution configuration does the following:

- Denies the redistribution of Type 2 external OSPFv3 routes with a tag set to five.
- Redistributes into IPv6 BGP all routes learned on the intf_ospf interface and sets the metric for such routes to 255.
- Redistributes into IPv6 BGP all other routes (those not processed by sequence 10 or 20) and sets the tag for such routes to eight.

IPv6 BGP Application Example

The following simple network using EBGP and IBGP will demonstrate some of the basic BGP setup commands discussed previously:

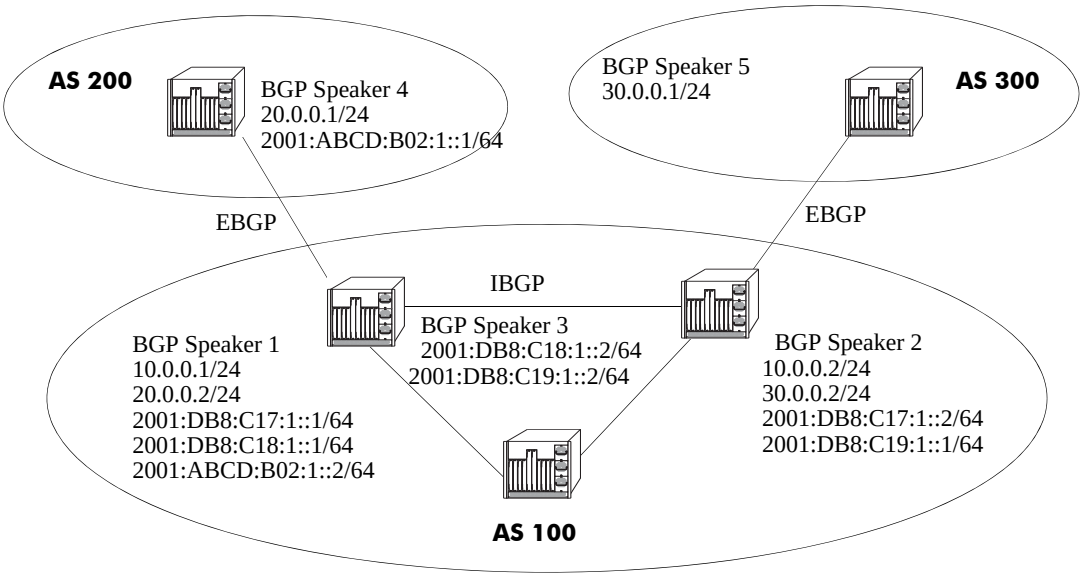


Figure 4-15 : IPv6 BGP Application Example

In the above network, Speakers 1, 2, and 3 are part of AS 100 and are fully meshed. Speaker 4 is in AS 200. Speaker 3 is part of a homogenous IPv6 network domain (i.e. pure IPv6 network), internal to AS 100. Speaker 5 in AS 300 is not aware of IPv6 capabilities.

AS 100

BGP Speaker 1

Assign the speaker to AS 100:

```
-> ip bgp autonomous-system 100
```

Enable IPv6 BGP unicast:

```
-> ipv6 bgp unicast
```

Peer with the other speakers in AS 100 (for internal BGP, and to create a fully meshed BGP network):

```
-> ip interface Link_To_Speaker2 vlan 2
-> ip interface Link_To_Speaker2 address 10.0.0.1/24

-> ipv6 interface Link_To_Speaker2 vlan 2
-> ipv6 address 2001:DB8:C17:1::1/64 Link_To_Speaker2

-> ipv6 bgp neighbor 2001:DB8:C17:1::2
-> ipv6 bgp neighbor 2001:DB8:C17:1::2 remote-as 100
-> ipv6 bgp neighbor 2001:DB8:C17:1::2 activate-ipv6
-> ipv6 bgp neighbor 2001:DB8:C17:1::2 ipv4-nexthop 10.0.0.1
-> ipv6 bgp neighbor 2001:DB8:C17:1::2 admin-state enable

-> ipv6 interface Link_To_Speaker3 vlan 3
-> ipv6 address 2001:DB8:C18:1::1/64 Link_To_Speaker3
-> ipv6 bgp neighbor 2001:DB8:C18:1::2
```

```
-> ipv6 bgp neighbor 2001:DB8:C18:1::2 remote-as 100
-> ipv6 bgp neighbor 2001:DB8:C18:1::2 activate-ipv6
-> ipv6 bgp neighbor 2001:DB8:C18:1::2 admin-state enable
```

Peer with the external speaker in AS 200 using its IPv4 address and an IPv6 forwarding interface (for IPv6 traffic):

```
-> ip interface Link_To_AS200 vlan 4
-> ip interface Link_To_AS200 address 20.0.0.2/24

-> ipv6 interface Link_to_AS200 vlan 4
-> ipv6 address 2001:ABCD:B02:1::2/64 Link_to_AS200

-> ip bgp neighbor 20.0.0.1
-> ip bgp neighbor 20.0.0.1 remote-as 200
-> ip bgp neighbor 20.0.0.1 activate-ipv6
-> ip bgp neighbor 20.0.0.1 ipv6-nexthop 2001:ABCD:B02:1::2
-> ip bgp neighbor 20.0.0.1 admin-state enable
```

Administratively enable BGP:

```
-> ip bgp admin-state enable
```

BGP Speaker 2

Assign the speaker to AS 100:

```
-> ip bgp autonomous-system 100
```

Enable IPv6 BGP unicast:

```
-> ipv6 bgp unicast
```

Peer with the other speakers in AS 100 (for internal BGP, and to create a fully meshed BGP network):

```
-> ip interface Link_To_Speaker1 vlan 2
-> ip interface Link_To_Speaker1 address 10.0.0.2/24

-> ipv6 interface Link_To_Speaker1 vlan 2
-> ipv6 address 2001:DB8:C17:1::2/64 Link_To_Speaker1

-> ipv6 bgp neighbor 2001:DB8:C17:1::1
-> ipv6 bgp neighbor 2001:DB8:C17:1::1 remote-as 100
-> ipv6 bgp neighbor 2001:DB8:C17:1::1 activate-ipv6
-> ipv6 bgp neighbor 2001:DB8:C17:1::1 ipv4-nexthop 10.0.0.2
-> ipv6 bgp neighbor 2001:DB8:C17:1::1 admin-state enable

-> ipv6 interface Link_To_Speaker3 vlan 3
-> ipv6 address 2001:DB8:C19:1::1/64 Link_To_Speaker3

-> ipv6 bgp neighbor 2001:DB8:C19:1::2
-> ipv6 bgp neighbor 2001:DB8:C19:1::2 remote-as 100
-> ipv6 bgp neighbor 2001:DB8:C19:1::2 activate-ipv6
-> ipv6 bgp neighbor 2001:DB8:C19:1::2 admin-state enable
```

Peer with the external speaker in AS 300 using IPv4 address:

```
-> ip interface Link_To_AS300 vlan 4
-> ip interface Link_To_AS300 address 30.0.0.2/24

-> ip bgp neighbor 30.0.0.1
-> ip bgp neighbor 30.0.0.1 remote-as 300
-> ip bgp neighbor 30.0.0.1 admin-state enable
```

Administratively enable BGP:

```
-> ip bgp admin-state enable
```

BGP Speaker 3

Assign the speaker to AS 100:

```
-> ip bgp autonomous-system 100
```

Administratively disable IPv4 unicast, as this speaker is part of a homogeneous IPv6 domain:

```
-> no ip bgp unicast
```

Explicitly configure the router ID and the primary address of the speaker:

```
-> ip router router-id 10.0.0.3
-> ip router primary-address 10.0.0.3
```

Peer with the other speakers in AS 100 (for internal BGP, and to create a fully meshed BGP network):

```
-> ipv6 interface Link_To_Speaker1 vlan 2
-> ipv6 address 2001:DB8:C18:1::2/64 Link_To_Speaker1

-> ipv6 interface Link_To_Speaker2 vlan 3
-> ipv6 address 2001:DB8:C19:1::2/64 Link_To_Speaker2

-> ipv6 bgp neighbor address 2001:DB8:C18:1::1
-> ipv6 bgp neighbor address 2001:DB8:C18:1::1 remote-as 100
-> ipv6 bgp neighbor address 2001:DB8:C18:1::1 activate-ipv6
-> ipv6 bgp neighbor address 2001:DB8:C18:1::1 admin-state enable

-> ipv6 bgp neighbor address 2001:DB8:C19:1::1
-> ipv6 bgp neighbor address 2001:DB8:C19:1::1 remote-as 100
-> ipv6 bgp neighbor address 2001:DB8:C19:1::1 activate-ipv6
-> ipv6 bgp neighbor address 2001:DB8:C19:1::1 admin-state enable
```

Administratively enable BGP:

```
-> ip bgp admin-state enable
```

AS 200

BGP Speaker 4

Assign the speaker to AS 200:

```
-> ip bgp autonomous-system 200
```

Enable IPv6 BGP unicast:

```
-> ipv6 bgp unicast
```

Peer with the external speaker in AS 100 using its IPv4 address and an IPv6 forwarding interface (for IPv6 traffic):

```
-> ip interface Link_To_AS100 vlan 2
-> ip interface Link_To_AS100 address 20.0.0.1/24

-> ipv6 interface Link_to_AS100 vlan 2
-> ipv6 address 2001:ABCD:B02:1::1/64 Link_to_AS100

-> ip bgp neighbor 20.0.0.2
-> ip bgp neighbor 20.0.0.2 remote-as 100
-> ip bgp neighbor 20.0.0.2 activate-ipv6
-> ip bgp neighbor 20.0.0.2 ipv6-nexthop 2001:ABCD:B02:1::1
-> ip bgp neighbor 20.0.0.2 admin-state enable
```

Administratively enable BGP:

```
-> ip bgp admin-state enable
```

AS 300

BGP Speaker 5

Assign the speaker to AS 300:

```
-> ip bgp autonomous-system 300
```

Peer with the external speaker in AS 100 using its IPv4 address:

```
-> ip interface Link_To_AS100 vlan 2
-> ip interface Link_To_AS100 address 30.0.0.1/24

-> ip bgp neighbor 30.0.0.2
-> ip bgp neighbor 30.0.0.2 remote-as 100
-> ip bgp neighbor 30.0.0.2 admin-state enable
```

Administratively enable BGP:

```
-> ip bgp admin-state enable
```

Displaying IPv6 BGP Settings and Statistics

Use the show commands listed in the following table to display information about the current IPv6 BGP configuration and on IPv6 BGP statistics:

show ipv6 bgp network	Displays the status of all the IPv6 BGP networks or a specific IPv6 BGP network.
show ipv6 bgp path	Displays the known IPv6 BGP paths for all the routes or a specific route.
show ipv6 bgp routes	Displays the known IPv6 BGP routes.
show ipv6 bgp neighbors	Displays the configured IPv6 BGP peers.
show ipv6 bgp neighbors policy	Displays the timers for configured IPv6 BGP peers.
show ipv6 bgp neighbors statistics	Displays the neighbor statistics of the configured IPv6 BGP peers.
show ip bgp	Displays the current global settings for the local BGP speaker.
show ip bgp neighbors	Displays the configured IPv4 BGP peers.

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

Routing Policies

BGP selects routes for subsequent advertisement by applying policies available in a pre-configured local Policy Information database. This support of policy-based routing provides flexibility by applying policies based on the path (AS path list), community attributes (community lists), specific destinations (prefix lists and prefix6 lists), and so on.

You could also configure route maps to include all of the above in a single policy.

For BGP to do policy-based routing, each BGP peer needs to be tied to inbound and/or outbound policies (direction based on whether routes are being learned or advertised). Each one of the above policies can be assigned as an in-bound or out-bound policy for a peer.

First, you must create policies that match one of the specified criteria:

- AS Paths. An AS path list notes all of the ASs the route travels to reach its destination.
- Community List. Communities can affect route behavior based on the definition of the community.
- Prefix List. Prefix list policies filter IPv4 routes based on a specific IPv4 network address, or a range of IPv4 network addresses.
- Prefix6 List. Prefix6 list policies filter IPv6 routes based on a specific IPv6 network address, or a range of IPv6 network addresses.
- Route Map. Route map policies filter routes by amalgamating other policies into one policy.

Then you must assign these policies to a peer. Policies can be assigned to affect routes learned from the peer, routes being advertised to the peer, or both.

Note. Routing policies are supported for both IPv4 and IPv6 routes/peers.

Creating a Policy

There are four different types of policies that can be created using the CLI, as described above. Each policy has several steps that must be implemented for a complete policy to be constructed. Minimally, the policy must be named, defined, and enabled.

The following sections describe the process of creating the four types of policies.

Creating an AS Path Policy

AS path policies must be assigned a name and a regular expression. Regular expressions are a set of symbols and characters that represent an AS or part of an AS path. Regular expressions are fully described in “Regular Expressions” on page 4-12.

To create an AS path policy:

1 Use the **ip bgp policy aspath-list** command, with a regular expression and a name, as shown:

```
-> ip bgp policy aspath-list aspathfilter "^100 200$"
```

This AS path policy is called **aspathfilter**. The policy looks for routes with an AS path with the next hop AS 100, and originating from AS 200. Regular expressions must be enclosed by quotes.

2 Next, use the **ip bgp policy aspath-list action** command to set the policy action. The action of a policy is whether the route filtered by the policy is permitted or denied. Denied routes are not propagated by the BGP speaker, while permitted routes are. For example:

```
-> ip bgp policy aspath-list aspathfilter "^100 200$" action permit
```

The AS path policy **aspathfilter** now permits routes that match the regular expression ^100 200$. Regular expressions must be enclosed by quotes.

3 Optionally, you can set the priority for routes filtered by the policy using the **ip bgp policy aspath-list priority** command. Priority for policies indicates which policy should be applied first to routes. Routes with a high priority number are applied first. To set the policy priority, enter the policy name with the priority number, as shown:

```
-> ip bgp policy aspath-list aspathfilter "^100 200$" priority 10
```

The AS path policy **aspathfilter** now has a priority of 10. Regular expressions must be enclosed by quotes.

Creating a Community List Policy

Community list policies must be assigned a name and a community number. Predetermined communities are specified in RFC 1997.

To create a community policy:

1 Assign a name and community number to the policy using the **ip bgp policy community-list** command, as shown:

```
-> ip bgp policy community-list commfilter 600:1
```

The policy name is **commfilter** and it looks for routes in the community 600:1.

2 Set the policy action using the **ip bgp policy community-list action** command. The policy action either permits or denies routes that match the filter. Permitted routes are advertised, while denied routes are not. For example:

```
-> ip bgp policy community-list commfilter 600:1 action permit
```

The **commfilter** policy now permits routes in community 600:1 to be advertised.

3 Set the policy match type using the **ip bgp policy community-list match-type** command. The match type can be set to either **exact** or **occur**. An exact match only affects routes that are solely in the specified community, while an occur match indicates that the community can be anywhere in the community list. For example:

```
-> ip bgp policy community-list commfilter 600:1 match-type exact
```

Policy **commfilter** now looks for routes that only belong to the community 600:1.

4 Optionally, you can set the priority for routes filtered by the policy using the **ip bgp policy community-list priority** command. Priority for policies indicates which policy should be applied first to routes. Routes with a high priority number are applied first. To set the policy priority, enter the policy name with the priority number, as shown:

```
-> ip bgp policy community-list commfilter 500:1 priority 3
```

Policy **commfilter** now has a priority of 3.

Creating a Prefix List Policy

Prefix policies filter routes based on network addresses and their masks. You can also set prefix upper and lower limits to filter a range of network addresses.

To create a prefix list policy:

1 Name the policy and specify the IP network address and mask using the **ip bgp policy prefix-list** command, as shown:

```
-> ip bgp policy prefix-list prefixfilter 12.0.0.0 255.0.0.0
```

Prefix policy **prefixfilter** now filters routes that match the network address 12.0.0.0 with a mask of 255.0.0.0.

2 Set the policy action using the **ip bgp policy prefix-list action** command. The policy action either permits or denies routes that match the filter. Permitted routes are advertised, while denied routes are not. For example:

```
-> ip bgp policy prefix-list prefixfilter 12.0.0.0 255.0.0.0 action deny
```

Prefix policy **prefixfilter** now denies routes that match the network address 12.0.0.0 with a mask of 255.0.0.0.

3 Optionally, you can set a lower prefix limit on the addresses specified in the policy using the **ip bgp policy prefix-list ge** command. For example:

```
-> ip bgp policy prefix-list prefixfilter 14.0.0.0 255.0.0.0 ge 16
```

Prefix policy **prefixfilter** now denies routes after 14.0.0.0/16.

4 Optionally, you can set an upper prefix limit on the addresses specified in the policy using the **ip bgp policy prefix-list le** command. For example:

```
-> ip bgp policy prefix-list prefixfilter 14.0.0.0 255.0.0.0 le 24
```

Prefix policy **prefixfilter** now denies routes between 14.0.0.0/16 and 14.0.0.0/24

Creating a Prefix6 List Policy

Prefix6 policies filter routes based on IPv6 prefixes. You can also set prefix6 upper and lower limits for the minimum length of the prefix to be matched.

To create a prefix6 list policy:

1 Name the policy and specify the IPv6 address and the prefix length using the **ip bgp policy prefix6-list** command, as shown:

```
-> ip bgp policy prefix6-list uniqLocal FC00::/48
```

Prefix policy **uniqLocal** now filters routes that match the address FC00 with a prefix length of 48.

2 Set the policy action using the **ip bgp policy prefix6-list action** command. The policy action either permits or denies routes that match the filter. Permitted routes are advertised, while denied routes are not. For example:

```
-> ip bgp policy prefix6-list uniqLocal FC00::/48 action permit
```

Prefix policy **uniqLocal** now permits routes that match the address FC00 with a prefix length of 48.

3 Optionally, you can set a minimum length of the prefix to be matched in the policy using the **ip bgp policy prefix6-list ge** command. For example:

```
-> ip bgp policy prefix6-list uniqLocal FC00::/48 ge 48
```

Prefix policy **uniqLocal** now denies routes after FC00::/48

4 Optionally, you can set a maximum length of the prefix to be matched in the policy using the **ip bgp policy prefix6-list le** command. For example:

```
-> ip bgp policy prefix6-list uniqLocal FC00::/48 le 24
```

Prefix policy **uniqLocal** now denies routes between FC00::/48 and FC00::/24

Creating a Route Map Policy

Route map policies let you amalgamate the other policy types together, as well as add various other filters. For example, you could have a route map policy that includes both an AS path policy and a community policy.

To create a route map policy:

1 Name the route map policy and assign it a sequence number using the **ip bgp policy route-map** command. The sequence number allows for multiple instances of a policy, and orders the route map policies so that a lower sequence number is applied first. For example:

```
-> ip bgp policy route-map mapfilter 1
```

Route map policy **mapfilter** is now ready.

2 Set the policy action using the **ip bgp policy route-map action** command. The policy action either permits or denies routes that match the filter. Permitted routes are advertised, while denied routes are not. For example:

```
-> ip bgp policy route-map mapfilter 1 action deny
```

Prefix policy **mapfilter** now denies routes that are filtered.

3 Add various conditions to the route map policy. It is possible to add an AS path policy, a community policy, a prefix policy, a prefix6 policy, as well as indicate other variables such as local preference and weight. The following table displays a list of the commands that can be used to construct a route map policy:

Route Map Options	Command
Assigns an AS path matching list to the route map.	**ip bgp policy route-map aspath-list**
Configures the AS path prepend action to be taken when a match is found.	**ip bgp policy route-map asprepend**
Configures the action to be taken on the community attribute when a match is found.	**ip bgp policy route-map community**
Assigns a community matching list to the route map.	**ip bgp policy route-map community-list**
Configures the action to be taken for a community string when a match is found.	**ip bgp policy route-map community-mode**

Route Map Options	Command
Configures the local preference value for the route map.	**ip bgp policy route-map lpref**
Configures the action to be taken when setting local preference attribute for a local matching route.	**ip bgp policy route-map lpref-mode**
Configures a matching community primitive for the route map.	**ip bgp policy route-map match-community**
Configures a matching mask primitive in the route map.	**ip bgp policy route-map match-mask**
Configures a matching prefix primitive in the route map.	**ip bgp policy route-map match-prefix**
Configures a matching prefix6 primitive in the route map.	**ip bgp policy route-map match-prefix6**
Configures an AS path matching regular expression primitive in the route map.	**ip bgp policy route-map match-regexp**
Configures the Multi-Exit Discriminator (MED) value for a route map.	**ip bgp policy route-map med**
Configures the action to be taken when setting the Multi-Exit Discriminator (MED) attribute for a matching route.	**ip bgp policy route-map med-mode**
Configures the action to be taken on the origin attribute when a match is found.	**ip bgp policy route-map origin**
Assigns a prefix matching list to the route map.	**ip bgp policy route-map prefix-list**
Assigns a prefix6 matching list to the route map, which identifies the matching criteria list of IPv6 prefixes.	**ip bgp policy route-map prefix6-list**
Configures a BGP weight value to be assigned to inbound routes when a match is found.	**ip bgp policy route-map weight**
Configures the value to strip from the community attribute of the routes matched by this route map instance (sequence number).	**ip bgp policy route-map community-strip**

For example, to add AS path policy **aspathfilter** and community list policy **commfilter** to route map policy **mapfilter**, enter the following:

```
-> ip bgp policy route-map mapfilter 1 aspath-list aspathfilter
-> ip bgp policy route-map mapfilter 1 community-list commfilter
```

Note. Conditions added to a route map policy must have already been created using their respective policy commands. If you attempt to add non-existent policies to a route map policy, an error message is returned. Each component of a route map policy must be added using a separate CLI command as shown above.

Assigning a Policy to a Peer

Once policies have been created using the commands described above, the policies can be applied to **IPv4 and IPv6 routes** learned from a specific peer, or route advertisements to a specific peer.

The following table shows the list of commands that allow you to assign a policy to a peer:

BGP Attribute	Command
Assigns an inbound AS path list filter to a BGP peer.	**ip bgp neighbor in-aspathlist**
Assigns an inbound community list filter to a BGP peer.	**ip bgp neighbor in-communitylist**
Assigns an inbound prefix filter list to a BGP peer.	**ip bgp neighbor in-prefixlist**
Assigns an inbound prefix6 filter list to a BGP peer.	**ip bgp neighbor in-prefix6list**
Assigns an outbound AS path filter list to a BGP peer.	**ip bgp neighbor out-aspathlist**
Assigns an outbound community filter list to a BGP peer.	**ip bgp neighbor out-communitylist**
Assigns an outbound prefix filter list to a BGP peer.	**ip bgp neighbor out-prefixlist**
Assigns an outbound prefix6 filter list to a BGP peer.	**ip bgp neighbor out-prefix6list**
Assigns an inbound or outbound policy map to a BGP peer.	**ip bgp neighbor route-map**
Invokes an inbound or outbound policy re-configuration for a BGP peer.	**ip bgp neighbor clear soft**
Assigns an inbound AS path list filter to an IPv6 BGP peer.	**ipv6 bgp neighbor in-aspathlist**
Assigns an inbound community list filter to an IPv6 BGP peer.	**ipv6 bgp neighbor in-communitylist**
Assigns an inbound prefix filter list to an IPv6 BGP peer.	**ipv6 bgp neighbor in-prefixlist**
Assigns an inbound prefix6 filter list to an IPv6 BGP peer.	**ipv6 bgp neighbor in-prefix6list**
Assigns an outbound AS path filter list to an IPv6 BGP peer.	**ipv6 bgp neighbor out-aspathlist**
Assigns an outbound community filter list to an IPv6 BGP peer.	**ipv6 bgp neighbor out-communitylist**
Assigns an outbound prefix filter list to an IPv6 BGP peer.	**ipv6 bgp neighbor out-prefixlist**
Assigns an outbound prefix6 filter list to an IPv6 BGP peer.	**ipv6 bgp neighbor out-prefix6list**

BGP Attribute	Command
Assigns an inbound or outbound policy map to an IPv6 BGP peer.	**ipv6 bgp neighbor route-map**
Invokes an inbound or outbound policy reconfiguration for an IPv6 BGP peer.	**ipv6 bgp neighbor clear soft**

Policies that should affect routes learned from a peer use the **in-** prefix, and policies that affect routes being advertised to a peer use the **out-** prefix.

Assigning Inbound and Outbound Policies to an IPv4 Peer

The following sections describes assigning various policies to an IPv4 BGP peer.

Assigning In and Outbound AS Path Policies

AS path policies filter routes based on matches made to a set AS list in the route. An AS list is a list of all the ASs the route crosses until its destination. To filter routes learned from a peer by the AS list, enter the peer's IP address with the **ip bgp neighbor in-aspathlist** command as shown:

```
-> ip bgp neighbor 172.22.2.0 in-aspathlist aspathfilter
```

The AS path policy **aspathfilter** must be previously created using the **ip bgp policy aspath-list** command.

To attach the same policy on route advertisements to the peer, enter the peer IP address with the **ip bgp neighbor out-aspathlist** command, as shown:

```
-> ip bgp neighbor 172.22.2.0 out-aspathlist aspathfilter
```

Assigning In and Outbound Community List Policies

Community list policies filter routes based on matches made to a list of communities of which the route is a member. Communities group routes by attaching labels to them specifying a behavior (such as **no export**).

To filter routes learned from a peer by the community list, enter the peer's IP address with the **ip bgp neighbor in-communitylist** command as shown:

```
-> ip bgp neighbor 172.22.2.0 in-communitylist commlistfilter
```

The community list policy **commlistfilter** must be previously created using the **ip bgp policy community-list** command.

To assign the same policy to route advertisements to the peer, enter the peer IP address with the **ip bgp neighbor out-communitylist** command, as shown:

```
-> ip bgp neighbor 172.22.2.0 out-communitylist commlistfilter
```

Assigning In and Outbound Prefix List Policies

Prefix list policies filter routes based on a specific routing network, using an IP address or a series of IP addresses.

To filter routes learned from a peer by the prefix list, enter the peer's IP address with the **ip bgp neighbor in-prefixlist** command as shown:

```
-> ip bgp neighbor 172.22.2.0 in-prefixlist prefixfilter
```

The route map policy **prefixfilter** must be previously created using the **ip bgp policy prefix-list** command.

To assign the same policy to route advertisements to the peer, enter the peer IP address with the **ip bgp neighbor out-prefixlist** command, as shown:

```
-> ip bgp neighbor 172.22.2.0 out-prefixlist prefixfilter
```

Assigning In and Outbound Prefix6 List Policies

Prefix6 list policies filter IPv6 routes based on a specific IPv6 network address, or a range of IPv6 network addresses.

To filter routes learned from a peer by the prefix6 list, enter the peer's IP address with the **ip bgp neighbor in-prefix6list** command as shown:

```
-> ip bgp neighbor 172.22.2.0 in-prefix6list InboundPrefix6
```

The route map policy **InboundPrefix6** must be previously created using the **ip bgp policy prefix6-list** command.

To assign the same policy to route advertisements to the peer, enter the peer IP address with the **ip bgp neighbor out-prefix6list** command, as shown:

```
-> ip bgp neighbor 172.22.2.0 out-prefix6list OutboundPrefix6
```

Assigning In and Outbound Route Map Policies

Route map policies filter routes combining routing criteria such as AS path, community, etc.

To filter routes learned from a peer by the route map, enter the peer's IP address with the **ip bgp neighbor route-map** command as shown:

```
-> ip bgp neighbor 172.22.2.0 route-map mapfilter in
```

The route map policy **mapfilter** must be previously created using the **ip bgp policy route-map** command.

To assign the same policy to route advertisements to the peer, enter the peer IP address with the **ip bgp neighbor route-map** command, as shown:

```
-> ip bgp neighbor 172.22.2.0 route-map mapfilter out
```

Reconfiguring Peer Policies

You can configure policies and assign these policies to a BGP peer, either to control in-bound routes or out-bound routes advertisement. Additionally, it is possible to change or modify these peer policies, after they are assigned to a peer.

Once the policies have been modified, they have to be re-applied to the peer. To re-apply the policies to only the peer under consideration, you can use the in-reconfigure and the out-reconfigure commands.

To reconfigure a peer's in policies, enter the peer's IP address with the **ip bgp neighbor clear soft** command as shown:

```
-> ip bgp neighbor 172.22.2.0 clear soft in
```

To reconfigure a peer's out policies, enter the peer IP address with the **ip bgp neighbor clear soft** command, as shown:

```
-> ip bgp neighbor 172.22.2.0 clear soft out
```

Assigning Inbound and Outbound Policies to an IPv6 Peer

The following sections describes assigning various policies to an IPv6 BGP peer.

Assigning In and Outbound AS Path Policy

AS path policies filter routes based on matches made to a set AS list in the route. An AS list is a list of all the ASs the route crosses until its destination. To filter routes learned from a peer by the AS list, enter the IPv6 address of the BGP peer with the **ipv6 bgp neighbor in-aspathlist** command as shown:

```
-> ipv6 bgp neighbor 2001::1 in-aspathlist InBoundAspath
```

The AS path policy **InBoundAspath** must be previously created using the **ip bgp policy aspath-list** command. Any inbound routes from the BGP peer must match this AS path filter before being accepted or passed to an inbound policy

To attach the same policy on route advertisements to the peer, enter the peer IP address with the **ipv6 bgp neighbor out-aspathlist** command, as shown:

```
-> ipv6 bgp neighbor 2001::1 out-aspathlist outBoundAspath
```

Assigning In and Outbound Community List Policy

Community list policies filter routes based on matches made to a list of communities of which the route is a member. Communities group routes by attaching labels to them specifying a behavior (such as **no export**).

To filter routes learned from a peer by the community list, enter the peer's IPv6 address with the **ipv6 bgp neighbor out-communitylist** command as shown:

```
-> ipv6 bgp neighbor 2001::1 in-communitylist InBoundCommlist
```

The community list policy **InBoundCommlist** must be previously created using the **ip bgp policy community-list** command. Any outbound routes from the BGP peer must match this community filter before being advertised or passed to inbound policy.

To assign the same policy to route advertisements to the peer, enter the peer IP address with the **ipv6 bgp neighbor out-communitylist** command, as shown:

```
-> ipv6 bgp neighbor 2001::1 out-communitylist OutboundCommList
```

Assigning In and Outbound Prefix List Policy

Prefix list policies filter routes based on a specific routing network, using an IP address or a series of IP addresses.

To filter routes learned from a peer by the prefix list, enter the peer's IPv6 address with the **ipv6 bgp neighbor in-prefixlist** command as shown:

```
-> ipv6 bgp neighbor 2001::1 in-prefixlist InBoundPrefix
```

The route map policy **InBoundPrefix** must be previously created using the **ip bgp policy prefix-list** command.

To assign the same policy to route advertisements to the peer, enter the peer IPv6 address with the **ipv6 bgp neighbor out-prefixlist** command, as shown:

```
-> ipv6 bgp neighbor 2001::1 out-prefixlist OutBoundPrefix
```

Assigning In and Outbound Prefix6 List Policy

Prefix6 list policies filter IPv6 routes based on a specific IPv6 network address, or a range of IPv6 network addresses.

To filter routes learned from a peer by the prefix6 list, enter the peer's IPv6 address with the **ipv6 bgp neighbor in-prefix6list** command as shown:

```
-> ipv6 bgp neighbor 2001::2 in-prefix6list InboundPrefix6
```

The route map policy **InboundPrefix6** must be previously created using the **ip bgp policy prefix6-list** command.

To assign the same policy to route advertisements to the peer, enter the peer IPv6 address with the **ipv6 bgp neighbor out-prefix6list** command, as shown:

```
-> ipv6 bgp neighbor 2001::2 out-prefix6list OutboundPrefix6
```

Assigning In and Outbound Route Map Policy

Route map policies filter routes combining routing criteria such as AS path, community, etc.

To filter routes learned from a peer by the route map, enter the peer's IPv6 address with the **ipv6 bgp neighbor route-map** command as shown:

```
-> ipv6 bgp neighbor 2001::1 route-map InboundRoute in
```

The route map policy **InboundRoute** must be previously created using the **ip bgp policy route-map** command.

To assign the same policy to route advertisements to the peer, enter the peer IPv6 address with the **ipv6 bgp neighbor route-map** command, as shown:

```
-> ipv6 bgp neighbor 2001::1 route-map OutboundRoute out
```

Reconfiguring IPv6 Peer Policies

You can configure policies and assign the policies to a BGP peer, either to control in-bound routes or out-bound routes advertisement. Additionally, it is possible to change or modify these peer policies, after they are assigned to a peer.

Once the policies have been modified, they have to be re-applied to the peer. To re-apply the policies to only the peer under consideration, you can use the in-reconfigure and the out-reconfigure commands.

To reconfigure a peer's in policies, enter the peer's IP address with the **ipv6 bgp neighbor clear soft** command as shown:

```
-> ipv6 bgp neighbor 2001::1 clear soft in
```

To reconfigure a peer's out policies, enter the peer IP address with the **ipv6 bgp neighbor clear soft** command, as shown:

```
-> ipv6 bgp neighbor 2001::1 clear soft out
```

Displaying Policies

The following commands are used to display the various policies configured on a BGP router:

show ip bgp policy aspath-list	Displays information on policies based on AS path criteria.
show ip bgp policy community-list	Displays information on policies based on community list criteria.
show ip bgp policy prefix-list	Displays information on policies based on route prefix criteria.
show ip bgp policy prefix6-list	Displays information on policies based on route prefix6 criteria.
show ip bgp policy route-map	Displays information on currently configured route maps.
show ip bgp path	Displays the AS Path details for a specific route.
show ipv6 bgp path	Displays the known IPv6 BGP paths for all the routes or a specific route.
show ip bgp neighbors policy	Displays the configured policies for the IPv4 peers or a specific peer.
show ip bgp policy route-map	Displays policy route map parameters.
show ipv6 bgp neighbors policy	Displays the configured policies for all IPv6 peers or for a specific IPv6 peer.

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

Generalized TTL Security Mechanism (GTSM) for BGP or eBGP Peer

GTSM for BGP peer provides a simple mechanism to prevent external BGP (eBGP) peer sessions from CPU-utilization based attacks. It also provides protection against spoofed control packets, minimizing the distance (number of hops) from which the attack can originate.

When GTSM is enabled for a peer, the TTL (IPv4) and hop limit (IPv6) field of BGP control packets sent to the peer is set to 255. The TTL is decremented for each hop on the packets path, the receiver checks the TTL count to determine the number of hops which the packet traversed. If the number of hops exceeds the maximum configured value, the packet is dropped.

Configuring GTSM for eBGP Peer

GTSM is configured by defining the maximum number of intermediate hops that BGP control traffic may take between the two peer endpoints. GTSM can be configured for BGP or eBGP peers using IPv4 and IPv6 addresses.

GTSM must be manually configured on all the participating switch in the peering session. When GTSM is enabled, eBGP multihop must be disabled or vise-versa. Attempting to configure GTSM when eBGP multihop is configured or vice-versa will display an error message.

To enable GTSM for IPv4 eBGP peer, use the **ip bgp neighbor ttl-security** command. The following example allows a maximum of six intermediate hops between the peers. Since the ttl-security is set to 6, only IP packets with a minimum TTL value of 249 (255-6=249) or greater is accepted. If the BGP neighbor is more than six hops away, the packet is dropped:

```
-> ip bgp neighbor 10.0.0.1 ttl-security 6
```

To enable GTSM for IPv6 eBGP peer, use the **ipv6 bgp neighbor ttl-security** command. For example:

```
-> ipv6 bgp neighbor 2001:db8::1 ttl-security 6
```

If the BGP peers are directly connected, then the ttl-security value must be set to zero. If the value of the received BGP control messages is less than 255, the packet is discarded.

Example for IPv4 eBGP peer:

```
-> ip bgp neighbor 10.0.0.1 ttl-security 0
```

Example for IPv6 eBGP peer:

```
-> ipv6 bgp neighbor 2001:db8::1 ttl-security 0
```

To disable GTSM for eBGP peer, use the no form of the command.

Example for IPv4 eBGP peer:

```
-> ip bgp neighbor 10.0.0.1 no ttl-security
```

Example for IPv6 eBGP peer:

```
-> ipv6 bgp neighbor 2001:db8::1 no ttl-security
```

Verifying the GTSM Configuration for eBGP Peer

To verify the GTSM configuration for eBGP peers using IPv4 and IPv6 address, use the s**how ip bgp neighbors** and **show ipv6 bgp neighbours** commands respectively.

An output field “Neighbor TTL security” will display the configured TTL value. If GTSM is disabled, the value is displayed as “none”.

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

5 Configuring Multicast Address Boundaries

Multicast boundaries confine scoped multicast addresses to a particular domain. Confining scoped addresses helps to ensure that multicast traffic passed within a multicast domain does not conflict with multicast users outside the domain.

In This Chapter

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

Configuration procedures described in this chapter include:

- Configuring multicast address boundaries—see page 5-6.
- Verifying the multicast address boundary configuration—see page 5-7.

For information about additional multicast routing commands, see the “Multicast Routing Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Quick Steps for Configuring Multicast Address Boundaries

Using Existing IP Interfaces

1 Before attempting to configure a multicast address boundary, be sure that you have manually loaded the multicast protocol software for your network (e.g., PIM or DVMRP). Otherwise, you will receive an error stating that "the specified application is not loaded." To manually load multicast protocol software, use the **ip load** command. For example:

```
-> ip load pim
```

2 Configure a multicast address boundary for a VLAN interface using the **ip mroute-boundary** command. Information must include the interface IP address, followed by the multicast boundary address and the corresponding subnet mask. For example:

```
-> ip mroute-boundary vlan-3 239.120.0.0 255.255.0.0
```

On New IP Interface

1 Be sure that you have loaded one of the dynamic routing features (e.g., PIM). Otherwise, you will receive an error stating that "the specified application is not loaded." To load a dynamic routing feature, use the **ip load** command. For example:

```
-> ip load pim
```

2 Create a new IP interface on an existing VLAN by specifying a valid IP address. For example:

```
-> ip interface vlan-2 address 178.14.1.43 vlan 3
```

The VLAN must already be created on the switch. For information about creating VLANs, see the "Configuring VLANs" chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

3 Configure a multicast address boundary on the IP interface. Information must include the IP address assigned at step 2, as well as a scoped multicast address and the corresponding subnet mask. For example:

```
-> ip mroute-boundary vlan-2 239.120.0.0 255.255.0.0
```

Note. *Optional.* To verify the multicast boundary configuration, enter the **show ip mroute-boundary** command. The display is similar to the one shown here:

```
-> show ip mroute-boundary
Interface Name Interface Address Boundary Address
--------------+-----------------+-------------------
vlan-2           178.14.1.43       239.120.0.0/16
```

For more information about this display, see the "Multicast Routing Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Multicast Address Boundaries Overview

Multicast Addresses and the IANA

The Internet Assigned Numbers Authority (IANA) regulates unique parameters for different types of network protocols. For example, the IANA regulates addresses for IP, DVMRP, PIM, PIM-SSM, etc., and also provides a range of administratively scoped multicast addresses. For more information, refer to the section below.

Administratively Scoped Multicast Addresses

Multicast addresses 239.0.0.0 through 239.255.255.255 have been reserved by the IANA as administratively scoped addresses for use in private multicast domains. These addresses cannot be used for any other protocol or network function. Because they are regulated by the IANA, these addresses can theoretically be used by network administrators without conflicting with networks outside of their multicast domains. However, to ensure that the addresses used in a private multicast domain do not conflict with other domains (e.g., within the company network or out on the Internet), multicast address boundaries must be configured.

Source-Specific Multicast Addresses

Multicast addresses 232.0.0.0 through 232.255.255.255 have been reserved by the Internet Assigned Numbers Authority (IANA) as source-specific multicast (SSM) destination addresses. Addresses within this range are reserved for use by source-specific applications and protocols (e.g., PIM-SSM) and cannot be used for any other functions or protocols.

Multicast Address Boundaries

Without multicast address boundaries, multicast traffic conflicts can occur between domains. For example, a multicast packet addressed to 239.140.120.10 from a device in one domain could “leak” into another domain. If the other domain contains a device attempting to send a separate multicast packet with the same address, a conflict may occur. A boundary is used to eliminate these conflicts by confining multicast traffic on an IP interface. When a boundary is set, multicast packets with a destination address within the specified boundary *will not* be forwarded on the interface.

The figure below provides an example of a multicast address boundary configured on an interface.

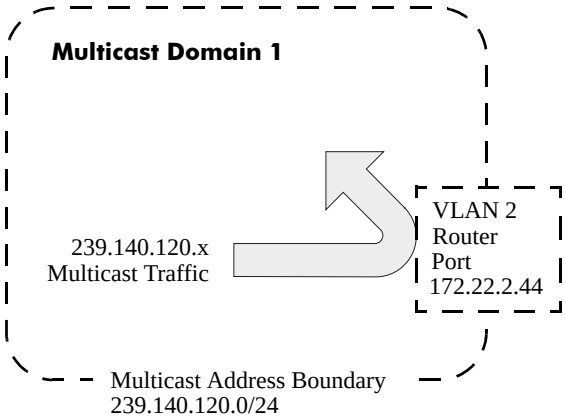


Figure 5-1 : Simple Multicast Address Boundary Example

An IP interface is configured on VLAN 2, with the IP address 172.22.2.44. The IP interface is also referred to as the router *interface*; the IP address serves as the identifier for the interface.

In this example, the multicast address boundary has been defined as 239.140.120.0. The mask value of 255.255.255.0 is shown in Classless Inter-Domain Routing (CIDR) prefix format as **/24**. This specifies that no multicast traffic addressed to multicast addresses 239.140.120.0 through 239.140.120.255 will be forwarded on interface 172.22.2.44.

Concurrent Multicast Addresses

Because multicast boundaries confine scoped multicast addresses to a particular domain, multicast addresses can be used concurrently in more than one region in the network. In other words, scoped multicast addresses can be reused throughout the network. This allows network administrators to conserve limited multicast address space. The figure below shows multicast addresses 239.140.120.0 through 239.140.120.255 being used by both Multicast Domain 1 and Multicast Domain 2.

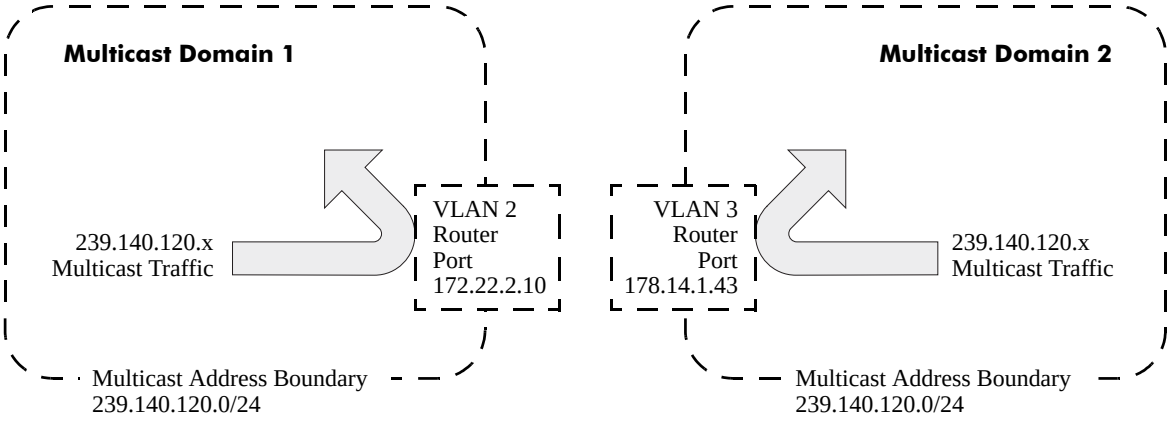


Figure 5-2 : Concurrent Multicast Addresses Example

Although the same block of multicast addresses—239.140.120.0 through 239.140.120.255—is being used in two different domains at once, multicast traffic from one domain cannot conflict with multicast traffic in the other domain because they are effectively confined by boundaries on their corresponding interfaces. In this case, the boundary 239.140.120.0/24 has been configured on interfaces 172.22.2.120 and 178.14.1.43.

Configuring Multicast Address Boundaries

Before configuring this feature, the multicast routing protocol (e.g., PIM or DVMRP) for your network must first be loaded to memory via the **ip load** command.

Basic Multicast Address Boundary Configuration

Configuring a multicast address boundary prevents multicast traffic that is addressed to a particular address or range of addresses from being forwarded on an interface. Boundaries may be configured in more than one region in the network.

The basic command for creating a multicast address boundary is:

ip mroute-boundary

The next section describes how to use this command.

Creating a Multicast Address Boundary

To create a multicast address boundary on an interface, enter the **ip mroute-boundary** command, with the interface IP address, the boundary address, and the corresponding mask. For example:

```
-> ip mroute-boundary vlan-2 239.120.0.0 255.255.0.0
```

The interface IP address must be a valid IP interface that has been assigned to an existing VLAN. For information about creating VLANs and assigning IP interfaces, see the “Configuring VLANs” chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

The boundary address must be an administratively-scoped multicast address from 239.0.0.0 to 239.255.255.255.

Deleting a Multicast Address Boundary

To delete a multicast address boundary from an interface, enter the **no ip mroute-boundary** command, with the interface IP address, the boundary address, and the corresponding mask. For example:

```
-> no ip mroute-boundary vlan-2 239.120.0.0 255.255.0.0
```

Verifying the Multicast Address Boundary Configuration

A summary of the show commands used for verifying the multicast address boundary configuration is given here:

show ip mroute-boundary	Displays scoped multicast address boundaries for the switch's router interfaces.

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

Application Example for Configuring Multicast Address Boundaries

This section illustrates multicast address boundary configuration for a simple multicast network. The network consists of a core switch with a backbone connection to the Internet. The core switch is given a boundary of 239.0.0.0/8. This is the broadest boundary, keeping all multicast traffic addressed to 239.0.0.0 through 239.255.255.255 from leaving the company network.

The core switch is connected to two wiring closet switches. The wiring closet switches serve the Human Resources and Training network domains. A boundary of 239.188.0.0/16 is created for both the Human Resources and Training domains. No multicast traffic within the range of 239.188.0.0 through 239.188.255.255 is permitted to leave either domain. This allows multicast addresses within the range to be used simultaneously in both domains without conflict.

Note. For a diagram showing this sample network with the multicast address boundaries described above, refer to page 5-10.

---

1 Verify that either PIM or DVMRP is loaded on the switch. Refer to the “Configuring PIM” or “Configuring DVMRP” chapters in the *OmniSwitch AOS Release 8 Advanced Routing Configuration Guide* for more information.

2 Create a VLAN on the core switch. For example:

```
-> vlan 2
```

3 Next, create an IP interface on the VLAN. The IP interface serves as the interface identifier on which the boundary will be created. To create a IP interface, use the **ip interface** command. For example:

```
-> ip interface vlan-2 address 178.10.1.1 vlan 2
```

4 You are now ready to create a boundary on the core switch’s router interface. For this example, the broadest possible boundary, 239.0.0.0, will be configured on the interface. This boundary will keep all traffic addressed to multicast addresses 239.0.0.0 through 239.255.255.255 from being forwarded on the interface. To assign the boundary, use the **ip mroute-boundary** command. For example:

```
-> ip mroute-boundary vlan-2 239.0.0.0 255.0.0.0
```

Note that the command includes the interface IP address (178.10.1.1), along with the multicast address boundary (239.0.0.0) and the corresponding subnet mask (255.0.0.0).

5 Verify your changes using the **show ip mroute-boundary** command:

```
-> show ip mroute-boundary
Interface Name Interface Address Boundary Address
--------------+-----------------+-------------------
vlan-2           178.10.1.1        239.0.0.0/8
```

The correct multicast address boundary of 239.0.0.0 is shown on VLAN 2. (VLAN 2 is displayed in the table because it contains the IP interface on which the boundary was configured. In this case, that IP interface is 178.10.1.1.) In addition, the subnet mask has been translated into the CIDR prefix length of **/8**.

The figure below illustrates the multicast address boundary as currently configured.

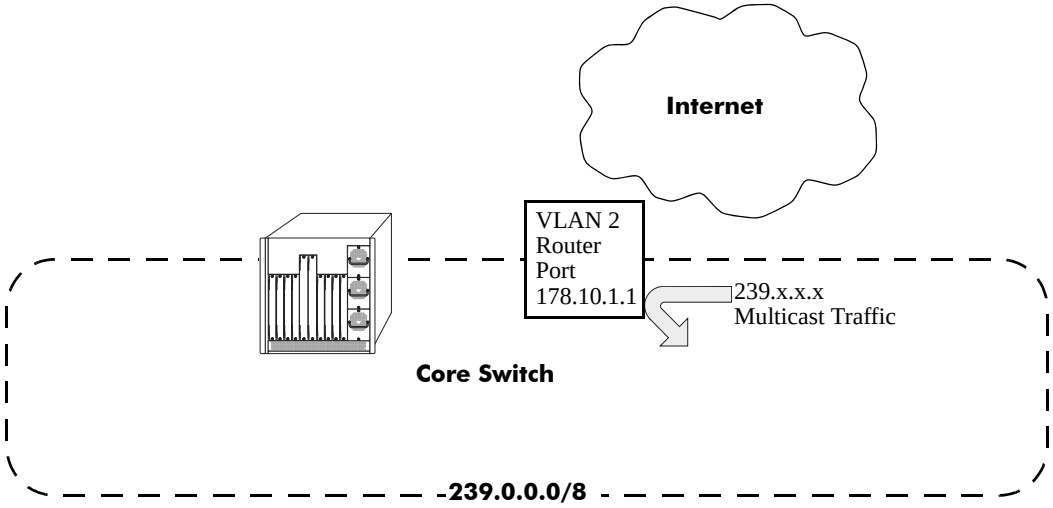


Figure 5-3 : Network with a Single Multicast Address Boundary

All multicast traffic ranging from 239.0.0.0 through 239.255.255.255 is blocked and cannot be forwarded from switch's 178.10.1.1 router interface. As shown by the arrow, multicast traffic addressed to 239.x.x.x cannot leave the domain.

6 Next, create a VLAN on the wiring closet switch used for Human Resources. For example:

```
-> vlan 3
```

VLAN 3 is now used to define the Human Resources network domain.

7 Create an IP interface on VLAN 3. For example:

```
-> ip interface vlan-3 address 178.20.1.1 vlan 3
```

8 Assign a boundary on the switch's router interface. For this example, the interface is given the boundary 239.188.0.0/16. This boundary will keep all traffic addressed to multicast addresses 239.188.0.0 through 239.188.255.255 from being forwarded on the interface:

```
-> ip mroute-boundary vlan-3 239.188.0.0 255.255.0.0
```

The command syntax includes the interface IP address (178.20.1.1), along with the multicast address boundary (239.188.0.0) and the corresponding subnet mask (255.255.0.0).

9 Create a VLAN on the separate wiring closet switch used for Training. For example:

```
-> vlan 4
```

VLAN 4 is now used to define the Training network domain.

10 Create an IP interface on VLAN 4. For example:

```
-> ip interface vlan-4 address 178.30.1.1 vlan 4
```

11 Assign a boundary on the Training router interface. The interface is given the same boundary as Human Resources (i.e., 239.188.0.0/16).

```
-> ip mroute-boundary vlan-4 239.188.0.0 255.255.0.0
```

Because there is a boundary configured at each domain, multicast users in Human Resources can forward 239.188.x.x multicast traffic without conflicting with users in Training who are forwarding traffic with the same addresses. By allowing addresses to be used concurrently in more than one department, network administrators can conserve limited scoped multicast address space.

The figure below illustrates all configured multicast address boundaries for this network.

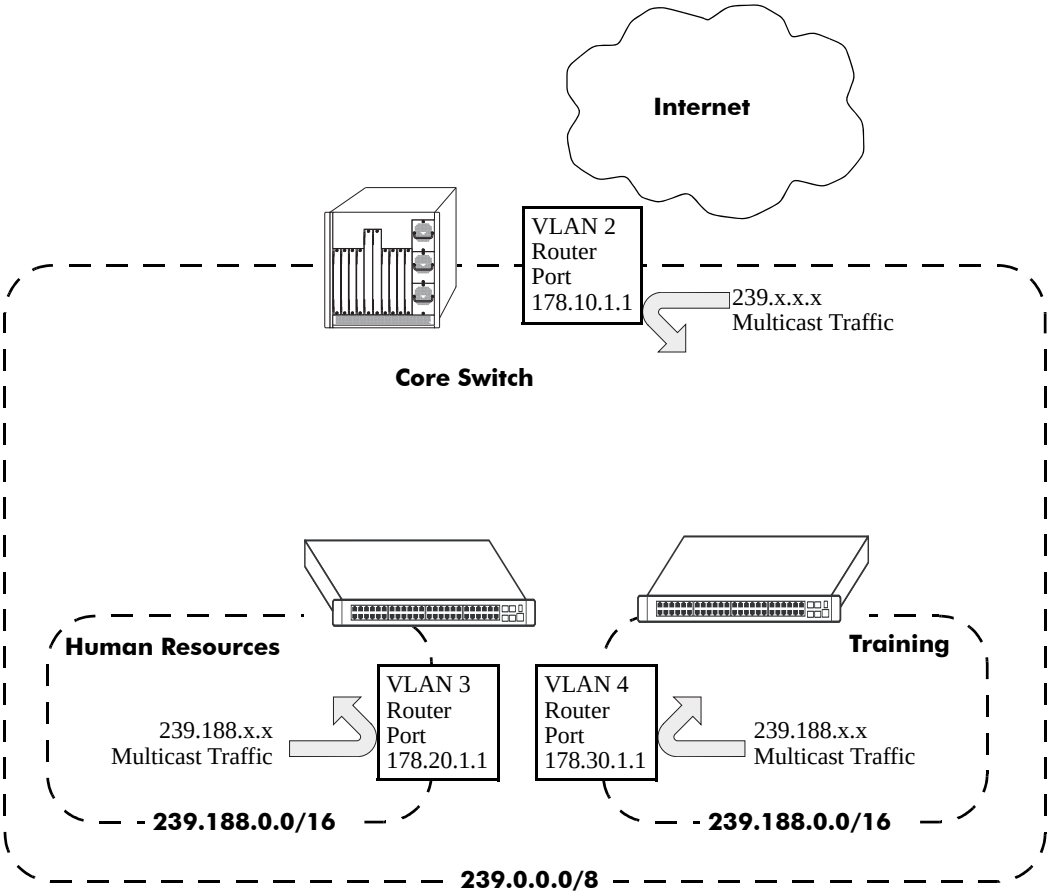


Figure 5-4 : Network with Multiple Multicast Addresses Boundaries

6 Configuring DVMRP

This chapter includes descriptions for Distance Vector Multicast Routing Protocol (DVMRP). DVMRP is a dense-mode multicast routing protocol. DVMRP, essentially a “broadcast and prune” routing protocol is designed to assist routers in propagating IP multicast traffic through a network.

In This Chapter

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

Configuration procedures described in this chapter include:

- Loading DVMRP into memory—see page 6-9.
- Enabling DVMRP—see page 6-11.
- Neighbor communications—see page 6-12.
- Routes—see page 6-13.
- Pruning—see page 6-14.
- Grafting—see page 6-16.
- Tunnels—see page 6-16.
- Verifying the DVMRP configuration—see page 6-17.

DVMRP Defaults

Parameter Description	Command	Default Value/Comments
DVMRP load status	**ip load dvmrp**	Unloaded
DVMRP status	**ip dvmrp admin-state**	Disabled
DVMRP interface status	**ip dvmrp interface**	Disabled
Flash update interval	**ip dvmrp flash-interval**	5 seconds
Graft retransmission timeout	**ip dvmrp graft-timeout**	5 seconds
Neighbor probe interval time	**ip dvmrp neighbor-interval**	10 seconds
Neighbor timeout	**ip dvmrp neighbor-timeout**	35 seconds
Prune lifetime	**ip dvmrp prune-lifetime**	7200 seconds
Prune retransmission timeout	**ip dvmrp prune-timeout**	30 seconds
Route report interval	**ip dvmrp report-interval**	60 seconds
Route hold-down time	**ip dvmrp route-holddown**	120 seconds
Route expiration timeout	**ip dvmrp route-timeout**	140 seconds
Interface distance metric	**ip dvmrp interface metric**	1
Subordinate neighbor status	**ip dvmrp subord-default**	true

Quick Steps for Configuring DVMRP

Note. DVMRP requires that IP Multicast Switching (IPMS) is enabled. IPMS is automatically enabled when a multicast routing protocol (either PIM or DVMRP) is enabled globally and on an interface *and* when the operational status of the interface is *up*. However, if you wish to manually enable IPMS on the switch, use the **ip multicast admin-state** command.

1 Manually load DVMRP into memory by entering the following command:

```
-> ip load dvmrp
```

2 Create a router port (i.e., *interface*) on an existing VLAN by specifying a valid IP address. To do this, use the **ip interface** command. For example:

```
-> ip interface vlan-2 address 178.14.1.43 vlan 2
```

3 Enable the DVMRP protocol on the interface via the **ip dvmrp interface** command. For example:

```
-> ip dvmrp interface vlan-2
```

4 Globally enable the DVMRP protocol by entering the following command:

```
-> ip dvmrp admin-state enable
```

5 Save your changes to the Working directory's **boot.cfg** file by entering the following command:

```
-> write memory
```

Once loaded and enabled, DVMRP is typically ready to use because its default values are appropriate for the majority of installations.

Note. *Optional*. To verify DVMRP interface status, enter the **show ip dvmrp interface** command. The display is similar to the one shown here:

```
 Address           Vlan   Metric  Admin-Status  Oper-Status
-----------------+------+--------+-------------+-------------
178.14.1.43        44       1       Enabled      Enabled
```

To verify the global DVMRP status, enter the **show ip dvmrp** command:

```
DVMRP Admin Status          = enabled,
Flash Interval              = 5,
Graft Timeout               = 5,
Neighbor Interval           = 10,
Neighbor Timeout            = 35,
Prune Lifetime              = 7200,
Prune Timeout               = 30,
Report Interval             = 60,
Route Holddown              = 120,
Route Timeout               = 140,
Subord Default              = true,
BFD status                  = disabled,
MBR Operational Status      = enabled,

Number of Routes            = 3,
Number of Reachable Routes  = 3
```

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

DVMRP Overview

Distance Vector Multicast Routing Protocol (DVMRP) Version 3 is a multicast routing protocol that enables routers to efficiently propagate IP multicast traffic through a network. Multicast traffic consists of a data stream that originates from a single source and is sent to hosts that have subscribed to that stream. Live video broadcasts, video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news services are examples of multicast traffic. Multicast traffic is distinguished from unicast traffic and broadcast traffic as follows:

- Unicast traffic is addressed to a single host.
- Broadcast traffic is transmitted to all hosts.
- Multicast traffic is transmitted to a subset of hosts (the hosts that have subscribed to the multicast data stream).

DVMRP is a distributed multicast routing protocol that dynamically generates per-source delivery trees based upon routing exchanges, using a technique called *Reverse Path Multicasting*. When a multicast source begins to transmit, the multicast data is flooded down the delivery tree to all points in the network. DVMRP then *prunes* (i.e., removes branches from) the delivery tree where the traffic is unwanted.

Pruning continues to occur as group membership changes or routers determine that no group members are present. This restricts the delivery trees to the minimum branches necessary to reach all group members, thus optimizing router performance. New branches can also be added to the delivery trees dynamically as new members join the multicast group. The addition of new branches is referred to as *grafting*.

Reverse Path Multicasting

DVMRP uses Internet Group Management Protocol (IGMP) messages to exchange the routing information needed to build per-source multicast delivery trees. Once built, packets follow a multicast delivery tree from the source to all members of the multicast group. Packets are replicated only at necessary branches in the delivery tree. The trees are calculated and updated dynamically to track the membership of individual groups.

When a packet arrives on an interface, the reverse path back to the source of the packet is determined by examining a DVMRP routing table of known source networks. If the packet arrived on an upstream interface that would be used to transmit packets back to the source, it is forwarded to the appropriate list of downstream interfaces. Otherwise, it is not on the optimal delivery tree and is discarded. In this way duplicate packets can be filtered when loops exist in the network topology.

Neighbor Discovery

DVMRP routers must maintain a database of DVMRP adjacencies with other DVMRP routers. A DVMRP router must be aware of its DVMRP neighbors on each interface. To gather this information, DVMRP routers use a neighbor discovery mechanism and periodically multicast DVMRP *Probe messages* to the All-DVMRP-Routers group address (224.0.0.4). Each Probe message includes a Neighbor List of DVMRP routers known to the transmitting router.

When a DVMRP router (let's call it "router B") receives a Probe (let's say from "router A"), it adds the IP address of router A to its own internal list of DVMRP neighbors on that interface. It then sends a Probe of its own with the IP address of router A included in the Probe's Neighbor List. When a DVMRP router receives a Probe with its own IP address included in the Neighbor List, the router knows that a two-way adjacency has been successfully formed between itself and the neighbor that sent the Probe.

Probes effectively serve three main purposes:

- Probes provide a mechanism for DVMRP routers to locate each other as described above.
- Probes provide a way for DVMRP routers to determine each others' capabilities. This is deduced from the major and minor version numbers in the Probe packet and directly from the capability flags in the Probe packet.
- Probes provide a keep-alive function in order to quickly detect neighbor loss.

A DVMRP router sends periodic *Route Report* messages to its DVMRP neighbors. A Route Report message contains the sender's current routing table, which contains entries that advertise a source network (with a mask) and a hop-count that is used as the routing metric.This routing information is used to build source distribution trees and to perform multicast forwarding. The DVMRP neighbor that advertises the route with the lowest metric will be used for forwarding. (In case of a tie, the DVMRP neighbor with the lowest IP address will be used.)

In DVMRPv3, a router will not accept a Route Report from another DVMRP router until it has established adjacency with that neighboring router.

Note. Older versions of DVMRP use Route Report messages to perform neighbor discovery rather than the Probe messages used in DVMRP Version 3.

Multicast Source Location, Route Report Messages, and Metrics

When an IP multicast packet is received by a router running DVMRP, it first looks up the source network in the DVMRP routing table. The interface that provides the best route back to the source of the packet is called the upstream interface. If the packet arrived on that upstream interface, then it is a candidate for forwarding to one or more downstream interfaces. If the packet did not arrive on that anticipated upstream interface, then it is discarded. This check is known as a *reverse path forwarding check* and is performed by all DVMRP routers.

Note. Under normal, stable DVMRP operation, packets would not arrive on the wrong interface because the upstream router would not forward the packet unless the downstream router poison-reversed the route in the first place (as explained below). However, there are cases—such as immediately after a network topology change—when DVMRP routing has not yet converged across all routers where this can occur. It can also occur when loops exist in the network topology.

In order to ensure that all DVMRP routers have a consistent view of the path back to a source, routing tables are propagated by all DVMRP routers in *Route Report messages*. Each router transmits a Route Report message at specified intervals. The Route Report message advertises the network numbers and masks of those interfaces to which the router is directly connected. It also relays the routes received from neighboring routers.

DVMRP requires an interface metric (i.e., a hop count) to be configured on all physical and tunnel interfaces. When a route is received from a neighboring router via a Route Report message, the metric of the interface over which the packet was received is added to the metric of the route being advertised. This adjusted metric is used when comparing metrics to determine the most efficient upstream interface.

Dependent Downstream Routers and Poison Reverse

In addition to providing a consistent view of source networks, the exchange of routes in DVMRP Route Report messages provides one other important feature. DVMRP uses the route exchange as a mechanism for upstream routers to determine if any downstream routers depend on them for forwarding packets from particular source networks.

DVMRP accomplishes this by using a technique called *poison reverse*. If a downstream router selects an upstream router as the best next hop to a particular source network, it indicates this by echoing back the route on the upstream interface with a metric equal to the original metric plus infinity. (DVMRP uses a metric of 32 as infinity.) When the upstream router receives the report and sees a metric that lies between infinity and twice infinity (that is, between 32 and 64), it adds the downstream router from which it received the report to a list of dependent routers for this source network.

The list of dependent routers per source network built by the poison reverse technique provides the foundation necessary to determine when it is appropriate to prune back the IP source-specific multicast trees.

Note. Poison reverse is used differently in DVMRP than in most unicast distance vector routing protocols (such as RIP), which use poison reverse to advertise that a particular route is unreachable.

Pruning Multicast Traffic Delivery

Initially, all interfaces with downstream-dependent neighbors are included in the downstream interface list and multicast traffic is flooded down the truncated broadcast tree to all possible receivers. This allows the downstream routers to be aware of traffic destined for a particular Source, Group (S, G) pair. The downstream routers then have the option to send prunes (and subsequent grafts) for this (S, G) pair as requirements change.

A DVMRP router will remove an interface from its forwarding list that has no group members associated with an IP multicast packet. If a router removes all of its downstream interfaces, it notifies the upstream router that it no longer wants traffic destined for that particular (S, G) pair. This is accomplished by sending a DVMRP Prune message upstream to the router expected to forward packets from that particular source.

A downstream router will inform an upstream router that it depends on the upstream router to receive packets from particular source networks by using the poison reverse technique during the exchange of Route Report messages. This method allows the upstream router to build a list of downstream routers on each interface that are dependent upon it for packets from a particular source. If the upstream router receives Prune messages from each one of the dependent downstream routers on an interface, then the upstream router can in turn remove this interface from its downstream interface list. If the upstream router is able to remove all of its downstream interfaces in this manner, it can then send a DVMRP Prune message to its upstream router. This continues until all unneeded branches are removed. Refer to "Pruning" on page 6-14 for more specific information on pruning.

Grafting Branches Back onto the Multicast Delivery Tree

A pruned branch will be automatically reattached to the multicast delivery tree when the prune times out. However, the graft mechanism provides a quicker method to reattach a pruned branch than waiting for the prune to time out. Without the graft mechanism, the join latency for new hosts in the group might be unacceptably great, because the prunes in the upstream routers would have to time out before multicast traffic could again begin to flow to the pruned branches. Depending on the number of routers along the pruned branch and the timeout values in use, several minutes might elapse before the host could begin to receive multicast traffic. By using a graft mechanism, DVMRP reduces the join latency to a few milliseconds.

The graft mechanism is made reliable through the use of Graft-Ack (Graft Acknowledgment) messages. A Graft-Ack message is returned by the upstream router in response to a Graft message. If the Graft-Ack message is not received, the downstream router will resend the Graft message. This prevents the loss of a Graft message due to congestion.

The **ip dvmrp graft-timeout** command enables you to set the Graft message retransmission value. This value defines the duration of time that the router will wait before retransmitting a Graft message if it has not received a Graft-Ack message. Refer to "Grafting" on page 6-16 for more information.

DVMRP Tunnels

Because not all IP routers support native multicast routing, DVMRP includes direct support for tunneling IP multicast packets through routers. Tunnel interfaces are used when routers incapable of supporting multicast traffic exist between DVMRP neighbors. In tunnel interfaces, IP multicast packets are encapsulated in unicast IP packets and addressed directly to the routers that do not support native multicast routing. DVMRP protocol messages (such as Route Reports, Probes for neighbor discovery, etc.) and multicast traffic are sent between tunnel endpoints using unicast, rather than multicast, packets.

Multicast data is encapsulated using a standard IP-IP encapsulation method. The unicast IP addresses of the tunnel endpoints are used as the source and destination IP addresses in the outer IP header. The inner IP header remains unchanged from the original multicast packet.

Configuring DVMRP

Before configuring DVMRP, consider the following:

- DVMRP requires that IP Multicast Switching (IPMS) is enabled. IPMS is automatically enabled when a multicast routing protocol (either PIM or DVMRP) is enabled globally and on an interface *and* when the operational status of the interface is up. However, if you wish to manually enable IPMS on the switch, use the **ip multicast admin-state** command.
- You can configure DVMRP parameters when the protocol is not running *as long as DVMRP is loaded into memory* (see "Loading DVMRP into Memory" below).
- The DVMRP parameters will *not* take effect until the protocol is enabled globally *and* on specific IP interfaces.

Enabling DVMRP on the Switch

Before running DVMRP, you must enable the protocol by completing the following steps:

- Loading DVMRP into memory
- Enabling DVMRP on desired IP interfaces
- Enabling DVMRP globally on the switch

Note. Once loaded and enabled, DVMRP is typically ready to use because its factory default values are appropriate for the majority of installations. Note, however, if neighbors in the DVMRP domain have difficulty handling large initial bursts of traffic, it is recommended that the subordinate neighbor status is changed to false. For more information on the subordinate neighbor status, refer to the **ip dvmrp subord-default** command in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

For information on completing these steps, refer to the sections below.

Loading DVMRP into Memory

You must load DVMRP into memory before you can begin configuring the protocol on the switch. If DVMRP is not loaded and you enter a configuration command, the following message displays:

```
ERROR: The specified application is not loaded
```

To dynamically load DVMRP into memory, enter the following command:

```
-> ip load dvmrp
```

Enabling DVMRP on a Specific Interface

Note. It does not matter whether DVMRP is first enabled globally or on specific interfaces. However, DVMRP will not run on an interface until it is enabled both globally and on the interface.

DVMRP must be enabled on an existing IP interface before any other interface-specific DVMRP command can be executed (e.g, the **ip dvmrp interface metric** command). The IP interface can be any interface that has been assigned to a valid IP address and an existing VLAN. For information about creating an IP interface, refer to the "Configuring IP" chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

To enable DVMRP on a specific interface, use the **ip dvmrp interface** command. The interface identifier used in the command syntax is the name of an existing IP interface. For example, the following command enables DVMRP on the "vlan-2" IP interface:

```
-> ip dvmrp interface vlan-2
```

Note. Only one multicast routing protocol is supported per interface. This means that you cannot enable both PIM and DVMRP on the same interface.

Disabling DVMRP on a Specific Interface

To disable DVMRP on a specific IP interface, use the **no ip dvmrp interface** command. Be sure to include the interface IP address. For example:

```
-> no ip dvmrp interface vlan-2
```

Specifying a Distance Metric on a Specific Interface

The **ip dvmrp interface metric** command enables you to specify the distance metric for an interface. DVMRP uses the metric value to determine the most cost-effective way of passing data. The higher an interface's metric value, the higher the cost of passing data over that interface. DVMRP will transmit data over the interface with the lowest available metric. Note that, just as in RIP, the metric of an incoming route advertisement is automatically incremented by the metric of the incoming interface.

To assign a distance metric to a specific interface, use the **ip dvmrp interface metric** command. The command syntax must include the interface as well as a distance metric value. For example:

```
-> ip dvmrp interface vlan-2 metric 10
```

Viewing DVMRP Status and Parameters for a Specific Interface

To view current DVMRP interfaces, including their operational status and assigned metrics, use the **show ip dvmrp interface** command. For example:

```
-> show ip dvmrp interface
Interface Name  Vlan   Metric   Admin-Status   Oper-Status
--------------+------+--------+-------------+--------------
vlan-2            2      1       Enabled      Enabled
```

Current assigned metric is shown as 1.

The corresponding interface is configured for DVMRP (i.e., it is DVMRP-enabled).

The interface is operationally down because there are no ports operationally up in VLAN 2.

Note. The **show ip dvmrp interface** command displays information for *all multicast-capable interfaces* (i.e. even interfaces where DVMRP might not be configured).

Globally Enabling DVMRP on the Switch

To globally enable DVMRP on the switch, enter the following command:

```
-> ip dvmrp admin-state enable
```

Globally Disabling DVMRP

The following command will globally disable DVMRP on the switch:

```
-> ip dvmrp admin-state disable
```

Checking the Current Global DVMRP Status

To view current global DVMRP enable/disable status, as well as additional global DVMRP settings, use the **show ip dvmrp** command. For example:

```
-> show ip dvmrp
DVMRP Admin Status          = enabled,
Flash Interval              = 5,
Graft Timeout               = 5,
Neighbor Interval           = 10,
Neighbor Timeout            = 35,
Prune Lifetime              = 7200,
Prune Timeout               = 30,
Report Interval             = 60,
Route Holddown              = 120,
Route Timeout               = 140,
Subord Default              = true,
BFD status                  = disabled,
MBR Operational Status      = enabled,

Number of Routes            = 20,
Number of Reachable Routes  = 18
```

Current global DVMRP status is shown as enabled.

Automatic Loading and Enabling of DVMRP Following a System Boot

If *any* DVMRP command is saved to the **boot.cfg** file in the post-boot running directory, DVMRP will be loaded into memory automatically. The post-boot running directory refers to the directory the switch will use as its running directory following the next system boot (i.e., Working or Certified). If the command syntax **ip dvmrp admin-state enable** is saved to the **boot.cfg** file in the post-boot running directory, DVMRP will be automatically loaded into memory *and* globally enabled following the next system boot. For detailed information on the Working and Certified directories and how they are used during system boot, see the "CMM Directory Management" chapter in the *OmniSwitch AOS Release 8 Switch Management Guide*.

Neighbor Communications

Probe messages are sent out periodically on all the DVMRP interfaces. However, only on the non-tunnel interfaces are they sent out to the multicast group address 224.0.0.4.

Note. Older versions of DVMRP use Route Report messages to perform neighbor discovery rather than the Probe messages used in DVMRP Version 3.

The **ip dvmrp neighbor-interval** command enables you to configure the interval, in seconds, at which Probe messages are transmitted. For example, to configure the Probe interval to ten seconds, enter the following command:

```
-> ip dvmrp neighbor-interval 10
```

The **ip dvmrp neighbor-timeout** command enables you to configure the number of seconds that the DVMRP router will wait for activity from a neighboring DVMRP router before assuming the neighbor is down. For example, to configure the neighbor timeout period to 35 seconds, enter the following command:

```
-> ip dvmrp neighbor-timeout 35
```

When the neighbor timeout expires and it is assumed that the neighbor is down, the following occurs:

- All routes learned from the neighbor are immediately placed in hold down.
- If the neighbor is considered to be the designated forwarder for any of the routes it is advertising, a new designated forwarder for each source network is selected.
- If the neighbor is upstream, any cache entries based upon this upstream neighbor are flushed.
- Any outstanding grafts awaiting acknowledgments from this neighbor are flushed.
- All downstream dependencies received from this neighbor are removed.

Set the default values for **ip dvmrp neighbor-interval** and **ip dvmrp neighbor-timeout**. This enables early detection of a lost neighbor yet provides tolerance for busy multicast routers. Both of these values must be coordinated between all DVMRP routers on a physical network segment.

Note. Current global DVMRP parameter values—including the **ip dvmrp neighbor-interval** value and the **ip dvmrp neighbor-timeout** value—can be viewed via the **show ip dvmrp** command. The DVMRP neighbor table can be viewed via the **show ip dvmrp neighbor** command.

Routes

In DVMRP, source network routing information is exchanged in the same basic manner as it is in RIP. That is to say, periodic Route Report messages are sent between DVMRP neighbors. A Route Report contains the sender's current routing table. The routing table contains entries that advertise a source network (with a mask) and a hop-count that is used as the routing metric. (The key difference between the way routing information is exchanged in DVMRP and in RIP is that DVMRP routes are advertised with a subnet mask, which makes DVMRP effectively a classless protocol.)

The routing information stored in a DVMRP routing table is separate from the unicast routing table and is used to build source distribution trees and to perform multicast forwarding (that is, Reverse Path Forwarding checks).

The **ip dvmrp report-interval** command enables you to specify the number of seconds between transmission of Route Report messages. For example, the following command specifies that a Route Report message be sent every 60 seconds:

```
-> ip dvmrp report-interval 60
```

The **ip dvmrp flash-interval** command enables you to specify the number of seconds between transmission of Routing Table Change messages. Routing Table Change messages are sent between transmissions of the complete routing tables contained in Route Report messages. For this reason, the Flash Interval value must be lower than the Route Report interval. For example:

```
-> ip dvmrp flash-interval 5
```

The **ip dvmrp route-timeout** command enables you to specify the route expiration timeout value. The route expiration timeout value determines the number of seconds before a route to an inactive network is aged out. For example, the following command specifies that the route to an inactive network age out in 140 seconds:

```
-> ip dvmrp route-timeout 140
```

The **ip dvmrp route-holddown** command enables you to specify the number of seconds that DVMRP routes are kept in a hold-down state. A hold-down state refers to the period of time that a route to an inactive network continues to be advertised as unreachable. When a route is deleted (because it expires, the neighbor it was learned from goes down, etc.) a router may be able to reach the source network described by the route through an alternate gateway. However, in the presence of complex topologies, often the alternate gateway may only be echoing back the same route learned via a different path. If this occurs, the route will continue to be propagated long after it is no longer valid.

In order to prevent this, it is common in distance vector protocols to continue to advertise a route that has been deleted with a metric of infinity for one or more report intervals. This is a hold-down. While it is in hold-down, a route must only be advertised with an infinity metric. The hold down period is usually two report intervals.

For example, the following command specifies that the route to an inactive network continue to be advertised for 120 seconds:

```
-> ip dvmrp route-holddown 120
```

Note. Current global DVMRP parameter values—including the **ip dvmrp report-interval**, **ip dvmrp flash-interval**, **ip dvmrp route-timeout**, and **ip dvmrp route-holddown** values—can be viewed via the **show ip dvmrp** command. The DVMRP routes that are being advertised to other routers can be viewed via the **show ip dvmrp route** command.

Pruning

DVMRP uses a flood-and-prune mechanism that starts by delivering multicast traffic to all routers in the network. This means that, initially, traffic is flooded down a multicast delivery tree. DVMRP routers then prune this flow where the traffic is unwanted. Routers that have no use for the traffic send DVMRP Prune messages up the delivery tree to stop the flow of unwanted multicast traffic, thus pruning the unwanted branches of the tree. After pruning, a source distribution tree for that specific source exists.

However, the source distribution tree that results from DVMRP pruning reverts back to the original delivery tree when the prunes time out. When a prune times out, traffic is again flooded down the branch.

The **ip dvmrp prune-lifetime** command sets the period of time that a prune will be in effect—essentially, the prune's lifetime. When the prune-lifetime period expires, the interface is joined back onto the multicast delivery tree. (If unwanted multicast traffic continues to arrive at the interface, the prune mechanism is reinitiated and the cycle continues.) For example, the following command sets a prune's lifetime to 7200 seconds:

```
-> ip dvmrp prune-lifetime 7200
```

Refer to "More About Prunes" below for further information on the **ip dvmrp prune-lifetime** command and how it affects the lifetime of prunes sent and, in some cases, received.

The **ip dvmrp prune-timeout** command sets the Prune packet retransmission interval. This is the duration of time that the router will wait before retransmitting a Prune message if it continues to receive unwanted multicast traffic. For example, the following command sets the Prune packet retransmission interval to forty seconds:

```
-> ip dvmrp prune-timeout 40
```

Note. Current global DVMRP parameter values—including the **ip dvmrp prune-lifetime** value and the **ip dvmrp prune-timeout** value—can be viewed via the **show ip dvmrp** command. Current DVMRP prunes can be viewed via the **show ip dvmrp prune** command.

More About Prunes

Prune-Lifetime Values in Sent Prune Packets

The default value is assigned to **ip dvmrp prune-lifetime**. On leaf routers (that is, routers that have no further downstream dependent routers), the value of **ip dvmrp prune-lifetime** is inserted into prune packets sent upstream as their lifetime value.

However, when a branch router (that is, a router that does have further downstream dependent routers) sends a prune upstream, the prune-lifetime value inserted into the prune packet is the smallest of the following values:

- the value of **ip dvmrp prune-lifetime** on the sending device
- the amount of lifetime that remains for each individual prune on the router's timer queue that was received for the pruned group. (When a prune is queued on the router's timer queue, its lifetime value decrements until the prune expires.)

As an example, let's say that the following situation exists on a branch router: **ip dvmrp prune-lifetime** is set to 7200 seconds and three prunes for the pruned group exist on the router's timer queue. These three prunes have remaining lifetimes of 7000 seconds, 5000 seconds, and 4500 seconds. When the branch router sends a prune upstream for this group, a prune-lifetime value of 4500 seconds will be inserted into the prune packet.

Prune-Lifetime Expiration Value

You can view the prunes that have been sent via the **show ip dvmrp prune** command. (However, note that this command does not display received prunes.) The expiration time displayed by the **show ip dvmrp prune** command is the earliest time that the router expects multicast traffic for the pruned group to start arriving. If the expiration time displays as **expired**, the prune has expired but no further multicast traffic has been received. The expiration value may be reset if multicast traffic is received and another prune was sent because no stations downstream want the traffic.

Received Prunes

When prune packets are received, a timer is set up on the receiving device that halts traffic sent to the pruned group on the neighbor that originated the prune. The timer value used is the prune-lifetime value found in the received prune packet. The setting of **ip dvmrp prune-lifetime** on the device that received the prune is not normally taken into consideration in this situation.

However, there are times when the setting of **ip dvmrp prune-lifetime** can affect the timeout value used for received prunes. This occurs if the setting of **ip dvmrp prune-lifetime** is modified after prunes have been received. If the new prune-lifetime value is less than the period of time a received prune has been on the router's timer queue, the router will treat the prune as if it just expired. This means that multicast traffic may flow to the neighbor even though the neighbor does not expect the prune to have expired.

Even in cases where modification of the **ip dvmrp prune-lifetime** setting does not cause the received prunes to expire earlier than specified by their internal prune-lifetime value, such modification will still cause the prune-lifetime value of received prunes to be adjusted to the new value. This means that received prunes may expire sooner or later than the neighbor expects.

Once the lifetime value of received prunes on the router's timer queue have been modified per the new setting of **ip dvmrp prune-lifetime**, all future incoming prunes will experience normal timer operation and the prune-lifetime value in the received prune packet will be used without modification. Outgoing prunes will use the new value of **ip dvmrp prune-lifetime**.

For the reasons explained, the value of **ip dvmrp prune-lifetime** should only be modified with caution.

Grafting

A pruned branch will be automatically reattached to the multicast delivery tree when the prune times out. However, the graft mechanism provides a quicker method to reattach a pruned branch than waiting for the prune to time out. As traffic is forwarded, routers that do not want multicast traffic send Prune messages to signal the upstream router to stop sending the traffic. If new IGMP membership requests are later received by the downstream router, the router can send Graft messages to the upstream router and wait for acknowledgment (a Graft Ack).

The **ip dvmrp graft-timeout** command enables you to set the Graft message retransmission value. This value defines the duration of time that the router will wait before retransmitting a Graft message if it has not received a Graft-Ack message acknowledging that a previously transmitted Graft message was received. For example, enter the following to set the Graft message retransmission value to 5 seconds:

```
-> ip dvmrp graft-timeout 5
```

Note. Current global DVMRP parameter values, including the **ip dvmrp graft-timeout** value, can be viewed via the **show ip dvmrp** command.

Tunnels

DVMRP networks may use DVMRP tunnels to interconnect two multicast-enabled networks across non-multicast networks. In a DVMRP tunnel, IP multicast packets are encapsulated in unicast IP packets so that the multicast traffic can traverse a non-multicast network.

The **ip interface tunnel** command enables you to add or delete an IP-IP tunnel and DVMRP can then be run over this tunnel. Any packets sent through the tunnel will be encapsulated in an outer IP header. For example, the following command would create a tunnel between local address 23.23.23.1 and remote address 155.2.2.2:

```
-> ip interface "tnl-1" tunnel source 23.23.23.1 destination 155.2.2.2
```

The local tunnel address must match an existing IP interface on a router that has been configured for DVMRP. The tunnel's remote address must be the IP address of the remote DVMRP router to which the tunnel is connected.

Important. DVMRP needs to be enabled on the IP interface of the source address of the tunnel and also on the configured tunnel interface.The tunnel will be operational only when the DVMRP interface is also operational. To enable DVMRP on an interface, use the **ip dvmrp interface** command. For more information, refer to "Enabling DVMRP on a Specific Interface" on page 6-10.

Note. Current DVMRP tunnels, including the tunnels' operational (OPER) status and TTL values, can be viewed via the **show ip dvmrp tunnel** command. The status of the DVMRP interface can be viewed via the **show ip dvmrp interface** command.

Verifying the DVMRP Configuration

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

show ip dvmrp	Displays global DVMRP parameters such as admin status, flash interval value, graft timeout value, neighbor interval value, subordinate neighbor status, number of routes, number of routes reachable, etc.
show ip dvmrp interface	Displays the DVMRP interface table, which lists all multicast-capable interfaces.
show ip dvmrp neighbor	Displays the DVMRP neighbor table, which lists adjacent DVMRP routers.
show ip dvmrp nexthop	Displays the DVMRP next hop entries table. The next hop entries table lists which VLANs will receive traffic forwarded from a designated multicast source. The table also lists whether a VLAN is considered a DVMRP branch or leaf for the multicast traffic (i.e., its *hop type*).
show ip dvmrp prune	Displays the prune table. Each entry in the prune table lists a pruned branch of the multicast delivery tree and includes the time interval remaining before the current prune state expires.
show ip dvmrp route	Displays the DVMRP routes that are being advertised to other routers in Route Report messages.
show ip dvmrp tunnel	Displays DVMRP tunnels. This command lists DVMRP tunnel interfaces, including both active and inactive tunnels.

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

7 Configuring PIM

Protocol-Independent Multicast (PIM) is an IP multicast routing protocol that uses routing information provided by unicast routing protocols such as RIP and OSPF. PIM is “protocol-independent” because it does not rely on any particular unicast routing protocol.

PIM-Sparse Mode (PIM-SM) contrasts with flood-and-prune dense mode multicast protocols, such as DVMRP and PIM-Dense Mode (PIM-DM), in that multicast forwarding in PIM-SM is initiated only via specific requests, referred to as *Join messages*. PIM-DM packets are transmitted on the same socket as PIM-SM packets, as both use the same protocol and message format. Unlike PIM-SM, in PIM-DM there are no periodic joins transmitted, only explicitly triggered prunes and grafts. In addition, there is no Rendezvous Point (RP) in PIM-DM. This release allows you to implement PIM in both the IPv4 and the IPv6 environments.

Note. This implementation of PIM includes support for Source-Specific Multicast (PIM-SSM). For more information on PIM-SSM support, refer to “PIM-SSM Support” on page 7-18.

In This Chapter

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

Configuration procedures described in this chapter include the following:

- Mapping an IPv6 multicast group to a PIM mode—see page 7-42.
- Configuring Candidate Rendezvous Points (C-RPs) in IPv6 PIM—see page 7-43.
- Configuring Candidate Bootstrap Routers (C-BSRs) in IPv6 PIM—see page 7-44.
- Configuring RP-switchover for IPv6 PIM—see page 7-47.
- Enabling PIM Join/Prune Message Packing for IPv6—see page 7-50.
- Verifying IPv6 PIM configuration—see page 7-51.

For detailed information about PIM commands, see the “PIM Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

PIM Defaults

The following table lists the defaults for the PIM configuration:

Parameter Description	Command	Default Value/Comments
PIM status	**ip load pim**	Disabled
PIM load status - sparse mode	**ip pim sparse admin-state**	Disabled
PIM load status - dense mode	**ip pim dense admin-state**	Disabled
Priority	**ip pim ssm group**	Disabled
Priority	**ip pim dense group**	Disabled
C-BSR mask length	**ip pim cbsr**	30 bits
Priority	**ip pim cbsr**	64
Static RP configuration	**ip pim static-rp**	Disabled
Priority	**ip pim candidate-rp**	192
C-RP advertisements	**ip pim candidate-rp**	60 seconds
RP threshold	**ip pim rp-threshold**	1
Keepalive timer	**ip pim keepalive-period**	210 seconds
Maximum RP routers allowed	**ip pim max-rps**	32
Probe timer	**ip pim probe-time**	5 seconds
Register checksum value	**ip pim register checksum**	header
Register suppression timer	**ip pim register-suppress-timeout**	60 seconds
Source, group data timeout	**ip pim keepalive-period**	210 seconds
Switchover to Shortest Path Tree (SPT)	**ip pim spt admin-state**	Enabled
Successive state refresh interval	**ip pim state-refresh-interval**	60 seconds
State refresh message limit	**ip pim state-refresh-limit**	0
State refresh ttl	**ip pim state-refresh-ttl**	16
Hello interval	**ip pim interface hello-interval**	30 seconds
Triggered hello	**ip pim interface triggered-hello**	5 seconds
Join/Prune interval	**ip pim interface joinprune-interval**	60 seconds
Hello holdtime	**ip pim interface hello-holdtime**	105 seconds
Join/Prune holdtime	**ip pim interface joinprune-holdtime**	210 seconds
Prune delay	**ip pim interface prune-delay**	500 milliseconds
Override interval	**ip pim interface override-interval**	2500 milliseconds
Designated Router Priority	**ip pim interface dr-priority**	1
Prune limit interval	**ip pim interface prune-limit-interval**	60 seconds
Graft retry interval	**ip pim interface graft-retry-interval**	3 seconds
Stub	**ip pim interface stub**	Disabled

Parameter Description	Command	Default Value/Comments
Neighbor loss notification interval	**ip pim neighbor-loss-notification-period**	0 seconds
Invalid register notification interval	**ip pim invalid-register-notification-period**	65535 seconds
RP mapping notification interval	**ip pim rp-mapping-notification-period**	65535 seconds
Invalid joinprune notification interval	**ip pim invalid-joinprune-notification-period**	65535 seconds
Interface election notification interval	**ip pim interface-election-notification-period**	65535 seconds
Join/Prune message packing	**ip pim joinprune-packing**	enable
Join/Prune MTU	**ip pim interface joinprune-mtu**	0 bytes
Join/Prune delay interval	**ip pim interface joinprune-delay**	0 milliseconds

IPv6 PIM Defaults

The following table lists the defaults for an IPv6 PIM configuration:

Parameter Description	Command	Default Value/Comments
PIM-SM status	**ipv6 pim sparse admin-state**	Disabled
PIM-DM status	**ipv6 pim dense admin-state**	Disabled
Priority	**ipv6 pim ssm group**	Disabled
Priority	**ipv6 pim dense group**	Disabled
Candidate-BSR	**ipv6 pim cbsr**	64 bits
Hash mask length	**ipv6 pim cbsr**	126
Static RP configuration	**ipv6 pim static-rp**	Disabled
Priority	**ipv6 pim candidate-rp**	192
C-RP advertisements	**ipv6 pim candidate-rp**	60 seconds
RP	**ipv6 pim rp-switchover**	Enabled
Switchover to Shortest Path Tree (SPT)	**ipv6 pim spt admin-state**	Enabled
Hello interval	**ipv6 pim interface hello-interval**	30 seconds
Triggered hello	**ipv6 pim interface triggered-hello**	5 seconds
Join Prune interval	**ipv6 pim interface joinprune-interval**	60 seconds
Hello holdtime	**ipv6 pim interface hello-holdtime**	105 seconds
Join Prune holdtime	**ipv6 pim interface joinprune-holdtime**	210 seconds
Prune delay	**ipv6 pim interface prune-delay**	500 milliseconds
Override interval	**ipv6 pim interface override-interval**	2500 milliseconds
Designated Router Priority	**ipv6 pim interface dr-priority**	1
Prune limit interval	**ipv6 pim interface prune-limit-interval**	60 seconds
Graft retry interval	**ipv6 pim interface graft-retry-interval**	3 seconds
Stub	**ipv6 pim interface stub**	Disabled
Join/Prune message packing	**ipv6 pim joinprune-packing**	enable
Join/Prune MTU	**ipv6 pim interface joinprune-mtu**	0 bytes
Join/Prune delay interval	**ipv6 pim interface joinprune-delay**	0 milliseconds

Quick Steps for Configuring PIM-DM

Note. PIM requires that IP Multicast Switching (IPMS) is enabled. IPMS is automatically enabled when a multicast routing protocol (either PIM or DVMRP) is enabled globally and on an interface *and* when the operational status of the interface is *up*. However, if you wish to manually enable IPMS on the switch, use the **ip multicast admin-state** command.

1 Manually load PIM into memory by entering the following command:

```
-> ip load pim
```

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

```
-> ip interface vlan-2 address 178.14.1.43 vlan 2
```

3 Enable PIM on the interface using the **ip pim interface** command. Note that the IP interface on which PIM is enabled must already exist in the switch configuration. For example:

```
-> ip pim interface vlan-2
```

4 Map the PIM-Dense Mode (DM) protocol for a multicast group using the **ip pim dense group** command. For example:

```
-> ip pim dense group 225.0.0.0/24
```

5 Globally enable the PIM protocol by entering the following command.

```
-> ip pim dense admin-state enable
```

6 Save your changes to the Working directory's **boot.cfg** file by entering the following command:

```
-> write memory
```

Note. *Optional*. To verify PIM interface status, enter the **show ip pim interface** command. The display is similar to the one shown here:

```
-> show ip pim interface
Total 1 Interfaces
Interface Name       IP Address      Designated      Hello    J/P      Oper
                                     Router          Interval Interval Status
-------------------+---------------+---------------+--------+--------+--------
tesvl                50.1.1.1        50.1.1.1        100      10       disabled
```

To verify global PIM status, enter the **show ip pim sparse** or **show ip pim dense** command. The display for sparse mode is similar to the one shown here:

```
-> show ip pim sparse
Status                      = enabled,
Keepalive Period            = 210,
Max RPs                     = 32,
Probe Time                  = 5,
Register Checksum           = header,
Register Suppress Timeout   = 60,
RP Threshold                = 1,
SPT Status                   = enabled,
```

The display for dense mode is similar to the one shown here:

```
-> show ip pim dense
Status                        = enabled,
Source Lifetime               = 210,
State Refresh Interval        = 60,
State Refresh Limit Interval = 0,
State Refresh TTL              = 16
```

(*additional table output not shown*)

For more information about these displays, see the "PIM Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

PIM Overview

Protocol-Independent Multicast (PIM) is an IP multicast routing protocol that uses routing information provided by unicast routing protocols such as RIP and OSPF. Note that PIM is not dependent on any particular unicast routing protocol.

Downstream routers must explicitly join PIM distribution trees in order to receive multicast streams on behalf of receivers or other downstream PIM routers. This paradigm of receiver-initiated forwarding makes PIM ideal for network environments where receiver groups are thinly populated and bandwidth conservation is a concern, such as in wide area networks (WANs).

Note. The OmniSwitch supports PIM-DM and PIM-SMv2 but is not compatible with PIM-SMv1.

PIM-Sparse Mode (PIM-SM)

Sparse mode PIM (PIM-SM) contrasts with flood-and-prune dense mode multicast protocols, such as DVMRP and PIM-Dense Mode (PIM-DM), in that multicast forwarding in PIM is initiated only via specific requests, referred to as *Join messages*.

The following sections provide basic descriptions for key components used when configuring a PIM-SM network. These components include the following:

- Rendezvous Points (RPs) and Candidate Rendezvous Points (C-RPs)
- Bootstrap Routers (BSRs) and Candidate Bootstrap Routers (C-BSRs)
- Designated Routers (DRs)
- Shared Trees, also referred to as Rendezvous Point Trees (RPTs)
- Avoiding Register Encapsulation

Rendezvous Points (RPs)

In PIM-SM, shared distribution trees are rooted at a common forwarding router, referred to as a Rendezvous Point (RP). The RP unencapsulates Register messages and forwards multicast packets natively down established distribution trees to receivers. The resulting topology is referred to as the RP Tree (RPT).

For an illustrated example of an RPT and the RP's role in a simple PIM-SM environment, refer to "Shared (or RP) Trees" on page 7-9.

Candidate Rendezvous Points (C-RPs)

A *Candidate* Rendezvous Point (C-RP) is a PIM-enabled router that sends periodic C-RP advertisements to the Bootstrap Router (BSR). When a BSR receives a C-RP advertisement, the BSR may include the C-RP in its RP-set. For more information on the BSR and RP-set, refer to page 7-9.

Bootstrap Routers (BSRs)

The role of a Bootstrap Router (BSR) is to keep routers in the network up to date on reachable C-RPs. The BSR's list of reachable C-RPs is also referred to as an *RP set*. There is only one BSR per PIM domain. This allows all PIM routers in the PIM domain to view the same RP set.

A C-RP periodically sends out messages, known as *C-RP advertisements*. When a BSR receives one of these advertisements, the associated C-RP is considered reachable (if it has a valid route). The BSR then periodically sends its RP set to neighboring routers in the form of a *Bootstrap message*.

Note. For information on viewing the current RP set, see page 7-27.

BSRs are elected from the Candidate Bootstrap Routers (C-BSRs) in the PIM domain. For information on C-BSRs, refer to the section below.

Candidate Bootstrap Routers (C-BSRs)

A *Candidate* Bootstrap Router (C-BSR) is a PIM-enabled router that is eligible for BSR status. To become a BSR, a C-BSR must become *elected*. A C-BSR sends Bootstrap messages to all neighboring routers. The messages include its IP address—which is used as an identifier—and its priority level. The C-BSR with the highest priority level is elected as the BSR by its neighboring routers. If two or more C-BSRs have the same priority value, the C-BSR with the highest IP address is elected as the BSR.

For information on configuring C-BSRs, including C-BSR priority levels, refer to "Candidate Bootstrap Routers (C-BSRs)" on page 7-25.

Designated Routers (DRs)

There is only one Designated Router (DR) used per LAN. When a DR receives multicast data from the source, the DR encapsulates the data packets into the Register messages, which are in turn sent to the RP. Downstream PIM routers express interest in receiving multicast streams on behalf of a host via explicit Join/Prune messages originating from the DR and directed to the RP.

The DR for a LAN is selected by an election process. This election process takes into account the DR priority of each PIM neighbor on the LAN. If multiple neighbors share the same DR priority, the neighbor with the highest IP address is elected. The **ip pim interface** command is used to specify the DR priority on a specific PIM-enabled interface. Note that the DR priority is taken into account only if all of the PIM neighbors on the LAN are using the DR priority option in their Hello packets.

For an illustrated example of the DR's role in a simple PIM environment, refer to "Shared (or RP) Trees" on page 7-9.

Shared (or RP) Trees

Shared distribution trees are also referred to as RP trees (or RPTs) because the routers in the distribution tree share a common Rendezvous Point (RP). The following diagrams illustrate a simple RP tree in a PIM-SM domain.

In this example, a multicast receiver (Receiver 1) uses IGMP to express interest in receiving multicast traffic destined for a particular multicast group. After getting the IGMP Join request, the receiver's Designated Router (DR) then passes on the request, in the form of a PIM *Join message*, to the RP.

Note. The Join message is known as a (*,G) join because it joins group G for all sources to that group.

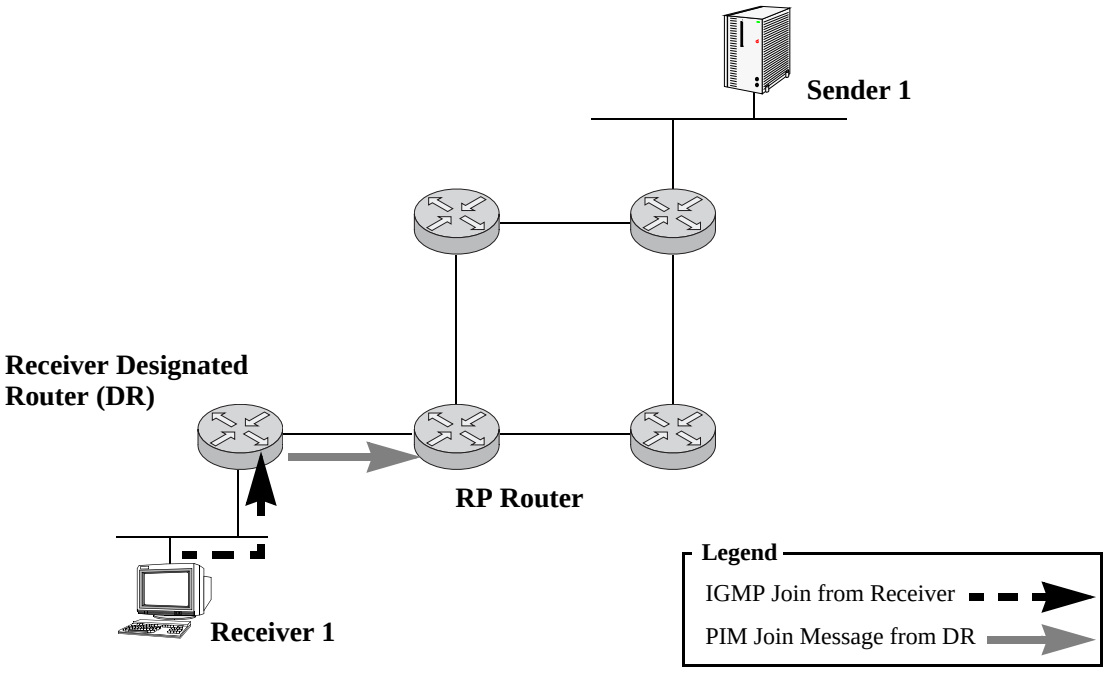


Figure 7-1 : RP tree in a PIM-SM domain.

Note. Depending on the network configuration, multiple routers may exist between the receiver's DR and the RP router. In this case, the (*, G) Join message travels hop-by-hop toward the RP. In each router along the way, the multicast tree state for group G is instantiated. These Join messages converge on the RP to form a distribution tree for group G that is rooted at the RP.

Sender 1 sends multicast data to its Designated Router (DR). The source DR then *unicast-encapsulates* the data into PIM-SM Register messages and sends them on to the RP.

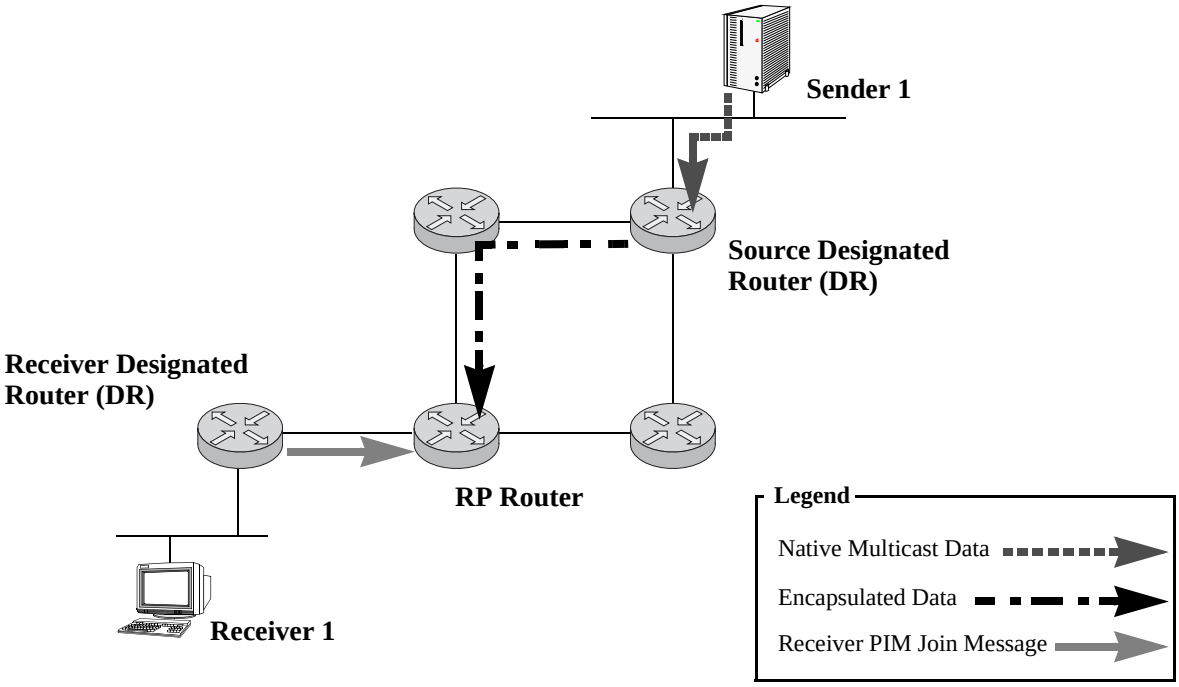


Figure 7-2 : RP tree in a PIM-SM domain - unicast-encapsulates data

Once the distribution tree for group G is learned at the RP, the encapsulated data being sent from the source DR are now unencapsulated at the RP and forwarded natively to the Receiver.

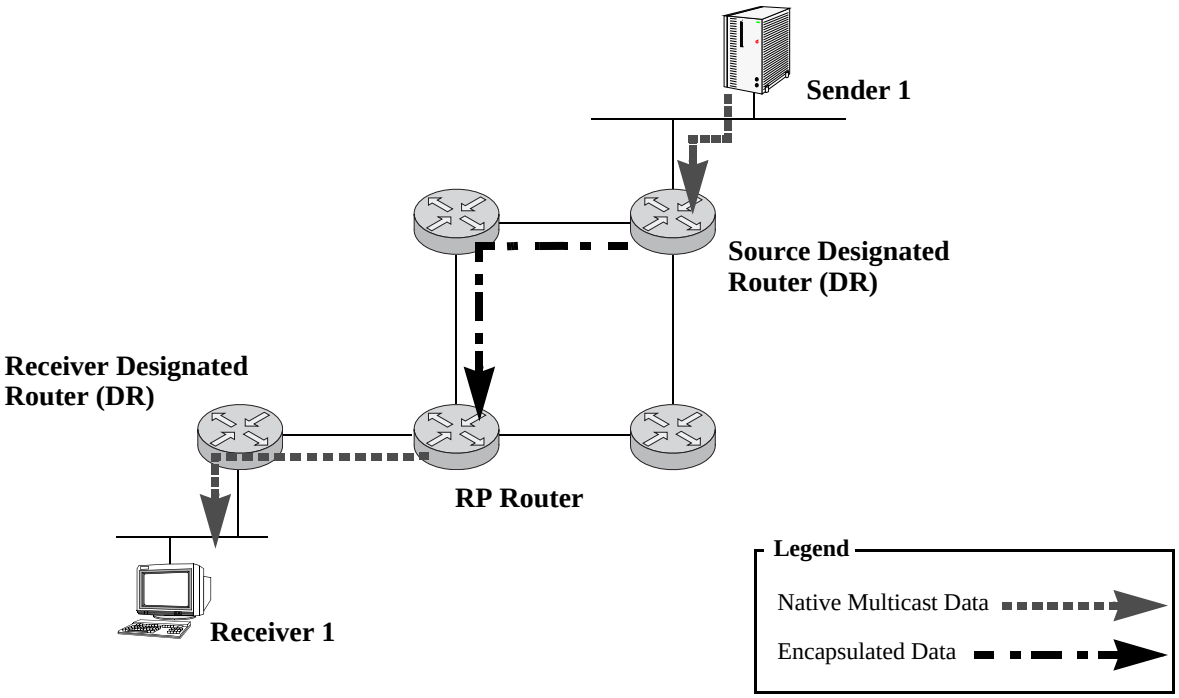


Figure 7-3 : RP tree in a PIM-SM domain - Unencapsulated data forwarded to Receiver

Avoiding Register Encapsulation

Switching to a Shortest Path Tree (SPT) topology allows PIM routers to avoid Register encapsulation of data packets that occurs in an RPT. Register encapsulation is inefficient for the following reasons:

- The encapsulation and unencapsulation of Register messages tax router resources. Hardware routing does not support encapsulation and unencapsulation.
- Register encapsulation may require that data travel unnecessarily over long distances. For example, data may have to travel "out of their way" to the RP before turning back down the shared tree in order to reach a receiver.

For some applications, this increased latency is undesirable. There are two methods for avoiding register encapsulation: RP initiation of (S, G) source-specific Join messages and switchover to a Shortest Path Tree (SPT). For more information, refer to the sections below.

PIM-Dense Mode (PIM-DM)

PIM-DM is a multicast routing protocol that defines a multicast routing algorithm for multicast groups densely distributed across a network. PIM-DM uses the underlying unicast routing information base to flood multicast datagrams to all multicast routers. Prune messages are used to prevent future messages from propagating to routers with no group membership information. It employs the same packet formats as PIM-SM.

PIM-DM assumes that when a multicast source starts sending, all downstream systems receive multicast datagrams. Multicast datagrams are initially flooded to all network areas. PIM-DM utilizes Reverse Path Forwarding to prevent looping of multicast datagrams while flooding. If some areas of the network do not have group members, PIM-DM will prune off the forwarding branch by instantiating the prune state.

PIM-DM differs from PIM-SM in two essential ways:

- There are no periodic joins transmitted, only explicitly triggered prunes and grafts.
- There is no Rendezvous Point (RP). This is particularly important in networks that cannot tolerate a single point of failure.

Note. A PIM router cannot differentiate a PIM-DM neighbor and a PIM-SM neighbor based on Hello messages, and PIM-DM is not intended to interact directly with a PIM-SM router.

RP Initiation of (S, G) Source-Specific Join Message

When the data rate at the Rendezvous Point (RP) exceeds the configured RP threshold value, the RP will initiate a (S, G) source-specific Join message toward the source.

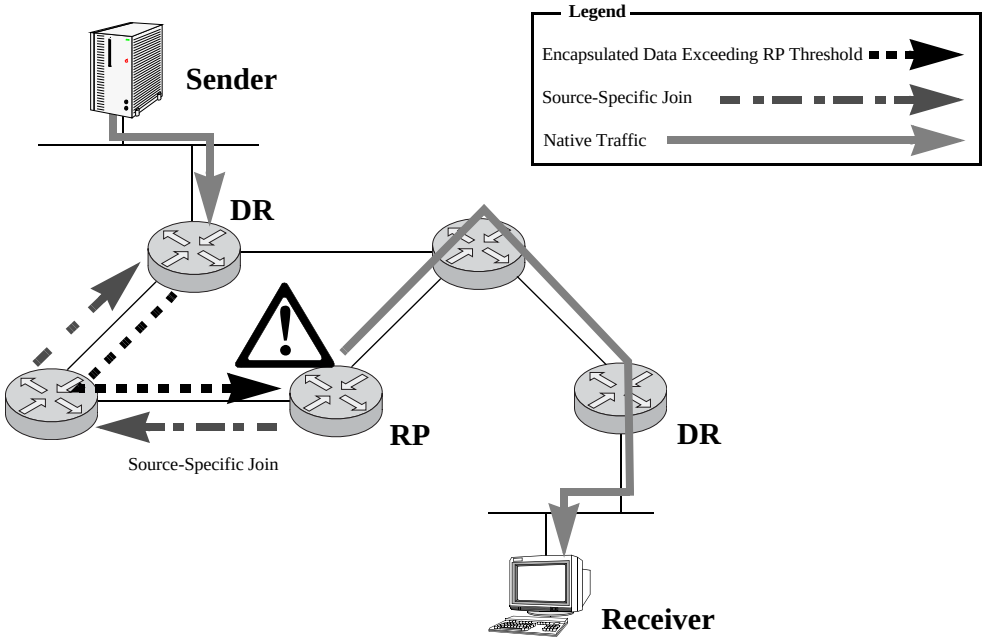


Figure 7-4 : RP Initiation of (S, G) Source-Specific Join Message (A)

Note. To configure the RP threshold value, use the **ip pim rp-threshold** command.

When the Sender's DR receives the (S,G) Join, it sends data natively as well. When these data packets arrive natively at the RP, the RP will be receiving *two copies* of each of these packets—one natively and one encapsulated. The RP drops the register-encapsulated packets and forwards only the native packets.

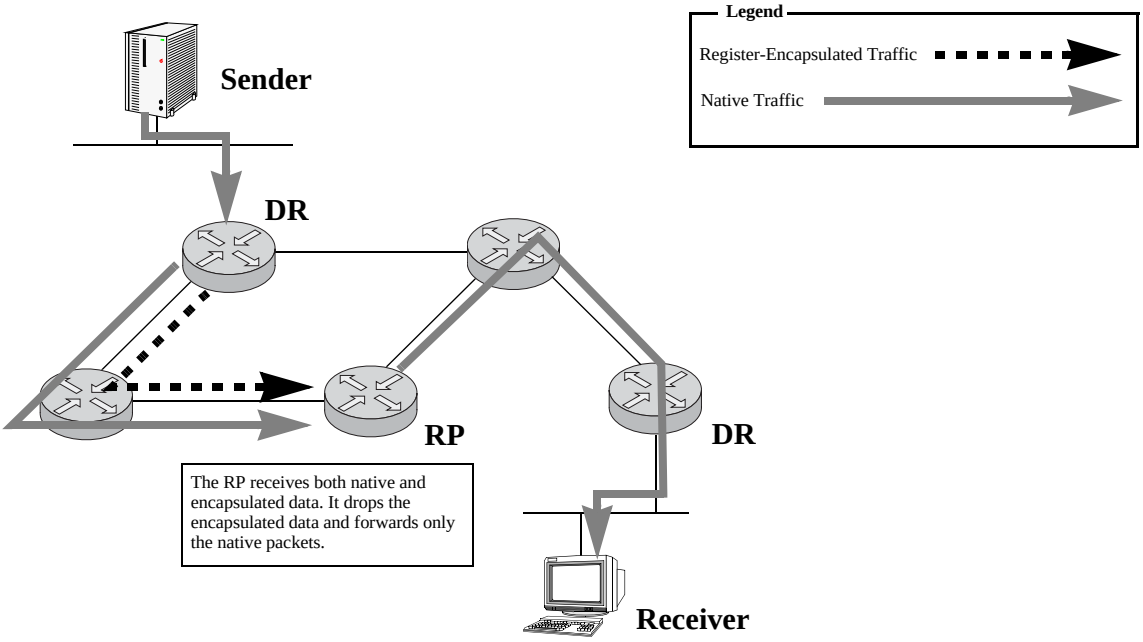


Figure 7-5 : RP Initiation of (S, G) Source-Specific Join Message (B)

A register-stop packet is sent back to the sender's DR to prevent the DR from unnecessarily encapsulating the packets. Once the register-encapsulated packets are discontinued, the packets flow natively from the sender to the RP—along the source-specific tree to the RP and, from there, along the shared tree to all receivers.

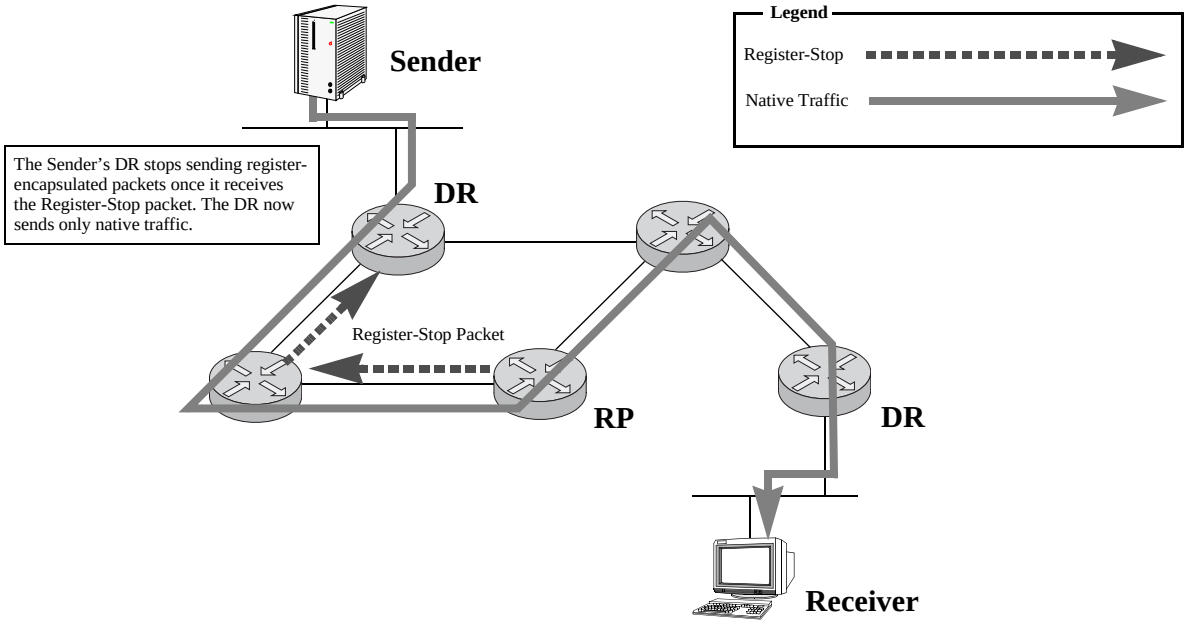


Figure 7-6 : RP Initiation of (S, G) Source-Specific Join Message (C)

Because packets are still forwarded along the shared tree from the RP to all of the receivers, this does not constitute a true Shortest Path Tree (SPT). For many receivers, the route via the RP may involve a significant detour when compared with the shortest path from the source to the receivers.

SPT Switchover

The last hop Designated Router (DR) initiates the switchover to a true Shortest Path Tree (SPT) once the DR receives the first multicast data packet. This method does not use any preconfigured thresholds, such as RP threshold (as described above). Instead, the switchover is initiated automatically, *as long as the SPT status is enabled on the switch*.

Important. SPT status must be enabled for SPT switchover to occur. If the SPT status is disabled, the SPT switchover will not occur. The SPT status is configured via the **ip pim spt admin-state** command. To view the current SPT status, use the **show ip pim sparse** command.

Upon receiving the first multicast data packet, the last hop DR issues a (S, G) source-specific Join message toward the source.

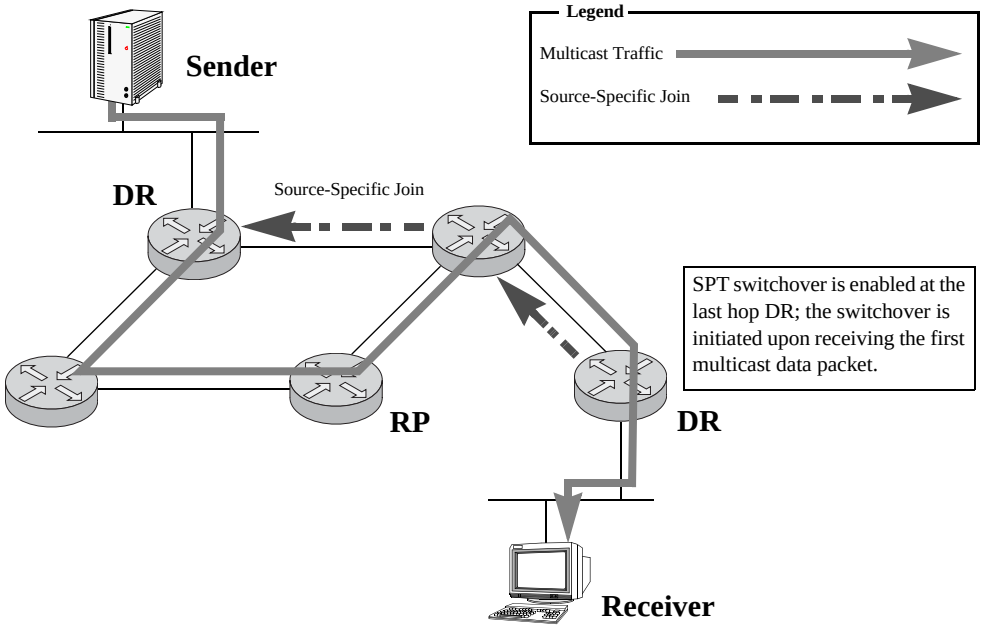


Figure 7-7 : SPT Switchover

Once the Sender's DR receives the (S,G) Join message, the DR sends the multicast packets natively along the Shortest Path Tree. At this point, Router X (the router shown between the Sender's DR and the Receiver's DR) will be receiving two copies of the multicast data—one from the SPT and one from the RPT. This router drops the packets arriving via the RP tree and forwards only those packets arriving via the SPT.

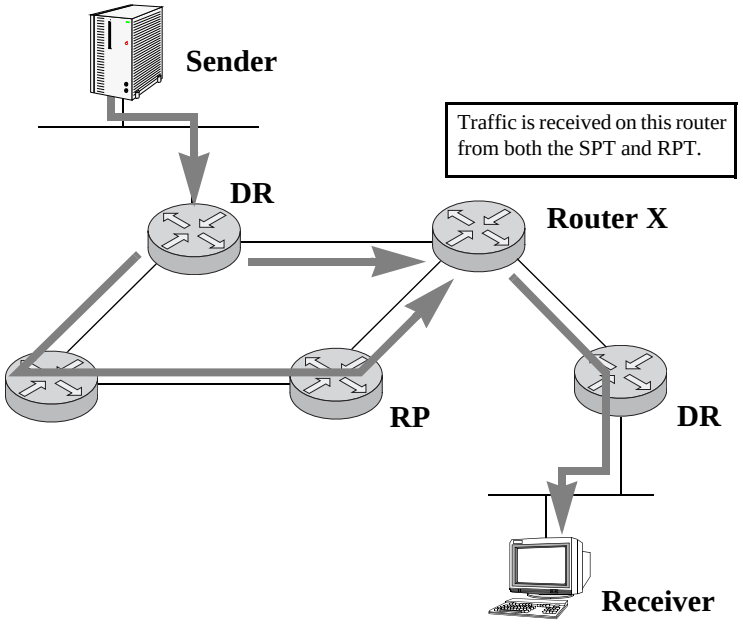


Figure 7-8 : SPT Switchover - Traffic received from both SPT and RPT

An (S, G, RPT) Prune message is sent toward the RP. As a result, traffic destined for this group from this particular source will no longer be forwarded along the RPT. The RP will still receive traffic from the Source. If there are no other routers wishing to receive data from the source, the RP will send an (S, G) Prune message toward the source to stop this unrequested traffic.

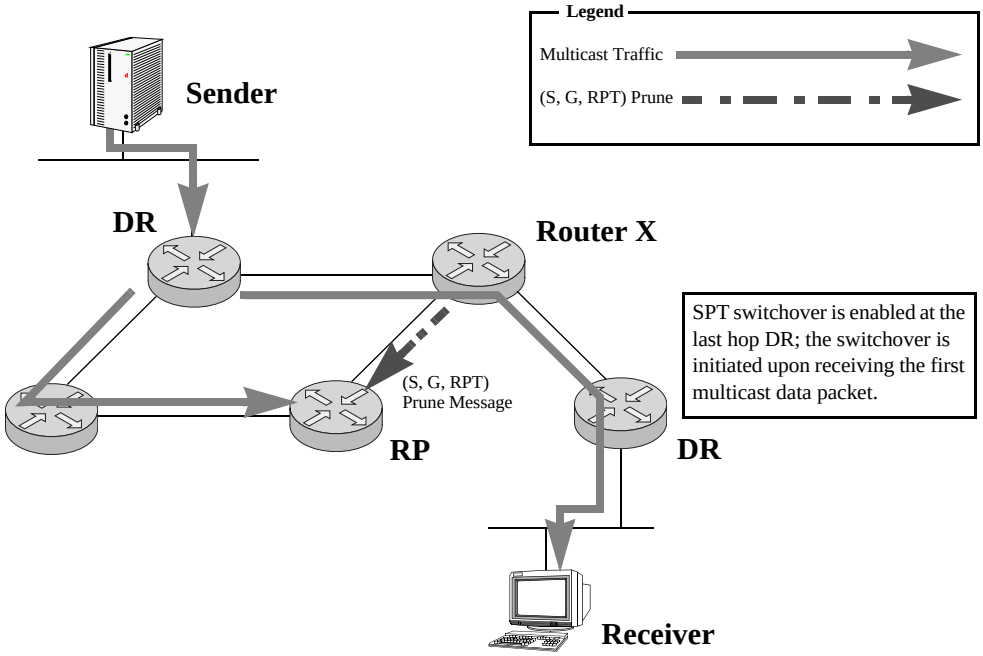


Figure 7-9 : SPT Switchover initiated upon receiving the first multicast data packet

The Receiver is now receiving multicast traffic along the Shortest Path Tree between the Receiver and the Source.

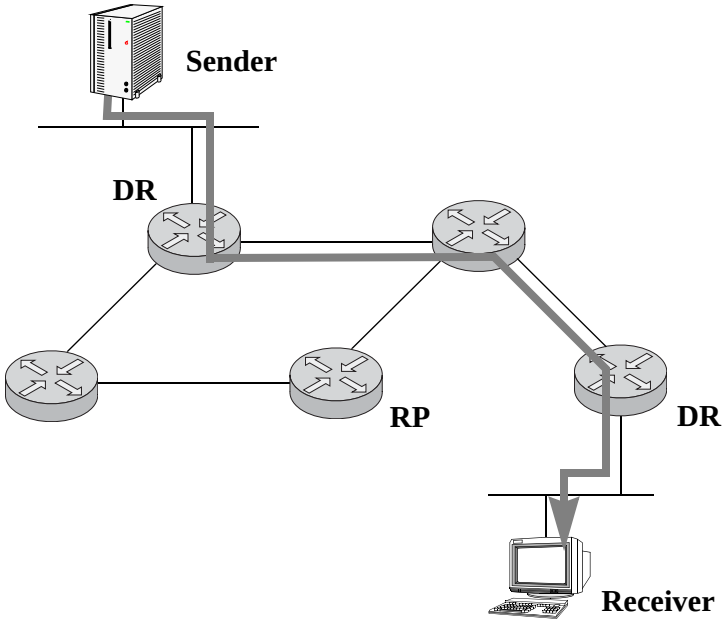


Figure 7-10 : Multicast traffic along the Shortest Path Tree

PIM-SSM Support

Protocol-Independent Multicast Source-Specific Multicast (PIM-SSM) is a highly-efficient extension of PIM. SSM, using an explicit channel subscription model, allows receivers to receive multicast traffic directly from the source; an RP tree model is not used. In other words, a Shortest Path Tree (SPT) between the receiver and the source is created without the use of a Rendezvous Point (RP).

PIM software supports Source-Specific Multicast. PIM-SSM is automatically enabled and operational as long as PIM is loaded (see page 7-6) and PIM sparse mode is enabled with the ssm group range defined (**ip pim ssm group**) and IGMPv3 source-specific joins are received within the SSM address range.

For detailed information on PIM-SSM and Source-Specific Multicast, refer to the IETF Internet Drafts draft-ietf-pim-sm-v2-new-05.txt and draft-ietf-ssm-arch-04.txt, as well as RFC 3569, “An Overview of Source-Specific Multicast (SSM).”

Note. For networks using IGMP proxy, be sure that the IGMP proxy version is set to Version 3. Otherwise, PIM-SSM will not function. For information on configuring the IGMP version, refer to the **ip multicast version** command.

Source-Specific Multicast Addresses

The multicast address range from 232.0.0.0 through 232.255.255.255 have been reserved by the Internet Assigned Numbers Authority (IANA) as Source-Specific Multicast (SSM) destination addresses. The PIM-Source-Specific Multicast (SSM) mode for the default SSM address range is not enabled automatically and needs to be configured manually to support SSM. Addresses within this range are reserved for use by source-specific applications and protocols (e.g., PIM-SSM). These addresses cannot be used for any other functions or protocols. However, you can also map additional multicast address ranges for the SSM group.

Configuring PIM

Enabling PIM on the Switch

Before running PIM, you must enable the protocol by completing the following steps:

- Verifying the software
- Loading PIM into memory
- Enabling PIM on desired IP interfaces
- Enabling PIM globally on the switch

Note. These steps are common for enabling PIM in the IPv4 as well as IPv6 environments.

For information on completing these steps, refer to the sections below.

Verifying the Software

To identify the current running directory (also referred to as *running configuration*), use the **show running-directory** command. For example:

```
-> show running-directory
CONFIGURATION STATUS
Running CMM              : PRIMARY,
CMM Mode                 : MONO CMM,
Current CMM Slot         : A,
Running configuration    : WORKING,
Certify/Restore Status   : CERTIFY NEEDED
SYNCHRONIZATION STATUS
Running Configuration    : SYNCHRONIZED,
NIs Reload On Takeover    : NONE
```

(additional table output not shown)

To view the software contents of the current running directory, use the **ls** command. If you are currently in the root flash, be sure to include the current running directory in the command line.

Loading PIM into Memory

You must load PIM into memory before you can begin configuring the protocol on the switch. If PIM is not loaded and you enter a configuration command, the following message displays:

```
ERROR: The specified application is not loaded
```

To dynamically load PIM into memory, enter the following command:

```
-> ip load pim
```

Enabling IPMS

PIM requires that IP Multicast Switching (IPMS) be enabled. IPMS is automatically enabled when a multicast routing protocol (either PIM or DVMRP) is enabled globally and on an interface *and* the operational status of the interface is up. If you wish to manually enable IPMS on the switch, use the **ip multicast admin-state** command.

Checking the Current IPMS Status

To view the current status of IPMS on the switch, use the **show ip multicast** command. For example:

```
-> show ip multicast
Status:                                          Enabled
Querying:                                        Disabled
Proxying:                                        Disabled
Spoofing:                                        Disabled
Zapping:                                         Disabled
Querier Forwarding:                              Disabled
Version:                                         2
Robustness:                                      2
Query Interval (seconds):                        125
Query Response Interval (tenths of seconds):  100
Last Member Query Interval(tenths of seconds):10
Unsolicited Report Interval(seconds):         1
Router Timeout (seconds):                        90
Source Timeout (seconds):                        30
```

Enabling PIM on a Specific Interface

PIM must be enabled on an interface using the **ip pim interface** command. An interface can be any IP router interface that has been assigned to an existing VLAN or a Shortest Path Bridging (SPB) service. For information on assigning a router interface to a VLAN or SPB service, refer to the "Configuring IP" chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

To enable PIM on a specific interface, use the **ip pim interface** command. The interface identifier used in the command syntax is the valid interface name of an existing VLAN IP router interface. For example:

```
-> ip pim interface vlan-2
```

Notes:

- Only one multicast routing protocol is supported per interface. This means that you cannot enable both DVMRP and PIM on the same interface.
- If the IP interface on which PIM is enabled is bound to an SPB service, then PIM can operate over an SPB L3 VPN in-line routing configuration (supported only on the OmniSwitch 9900).

Disabling PIM on a Specific Interface

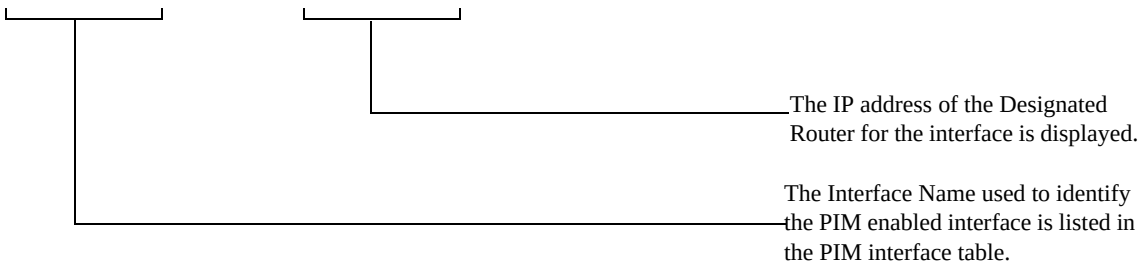
To disable PIM on a specific IP interface, use the **no ip pim interface** command. Be sure to include the name of the interface. For example:

```
-> no ip pim interface vlan-2
```

Viewing PIM Status and Parameters for a Specific Interface

To view the current PIM interface information—which includes IP addresses for PIM-enabled interfaces, Hello and Join/Prune intervals, and current operational status—use the **show ip pim interface** command. For example:

```
-> show ip pim interface
Total 1 Interfaces
Interface Name       IP Address      Designated      Hello    J/P      Oper
                                     Router          Interval Interval Status
--------------------+---------------+---------------+--------+--------+-------
tesvl                50.1.1.1        50.1.1.1        100      10       disabled
```



Enabling PIM Mode on the Switch

To globally enable PIM-Sparse Mode on the switch, use the **ip pim sparse admin-state** command. Enter the command syntax as shown below:

```
-> ip pim sparse admin-state enable
```

To globally enable PIM-Dense Mode on the switch, use the **ip pim dense admin-state** command. Enter the command syntax as shown below:

```
-> ip pim dense admin-state enable
```

Disabling PIM Mode on the Switch

To globally disable PIM-Sparse Mode on the switch, use the **ip pim sparse admin-state** command. Enter the command syntax as shown below:

```
-> ip pim sparse admin-state disable
```

To globally disable PIM-Dense Mode on the switch, use the **ip pim dense admin-state** command. Enter the command syntax as shown below:

```
-> ip pim dense admin-state disable
```

Checking the Current Global PIM Status

To view current global PIM enable/disable status, as well as additional global PIM settings, use the **show ip pim sparse** or **show ip pim dense** command. For example:

```
-> show ip pim sparse
Status                      = enabled,
Keepalive Period            = 210,
Max RPs                     = 32,
Probe Time                  = 5,
Register Checksum           = header,
Register Suppress Timeout   = 60,
RP Threshold                = 1,
SPT Status                   = enabled,

-> show ip pim dense
Status                          = enabled,
Source Lifetime                 = 210,
State Refresh Interval          = 60,
State Refresh Limit Interval    = 0,
State Refresh TTL                = 16
```

Mapping an IP Multicast Group to a PIM Mode

PIM mode is an attribute of the IP multicast group mapping and cannot be configured on an interface basis. The Dense mode or Source-Specific Multicast mode can be configured only on a multicast group basis.

Mapping an IP Multicast Group to PIM-DM

To statically map an IP multicast group(s) to PIM-Dense mode (DM), use the **ip pim dense group** command. For example:

```
-> ip pim dense group 225.0.0.0/24 priority 50
```

This command entry maps the multicast group 225.0.0.0/24 to PIM-DM and specifies the priority value to be used for the entry as 50. This priority specifies the preference value to be used for this static configuration and provides fine control over which configuration is overridden by this static configuration. Values may range from 0 to 128. If the priority option has been defined, a value of 65535 can be used to unset the priority.

You can also use the **override** parameter to specify whether or not this static configuration overrides the dynamically learned group mapping information for the specified group. As specifying the priority value obsoletes the **override** option, you can use only the **priority** parameter or the **override** parameter. By default, the **priority** option is not set and the **override** option is set to false.

Use the **no** form of this command to remove a static configuration of a dense mode group mapping.

```
-> no ip pim dense group 225.0.0.0/24
```

Mapping an IP Multicast Group to PIM-SSM

To statically map an IP multicast group(s) to PIM-Source-Specific Multicast mode (SSM), you can use the **ip pim ssm group** command. For example:

```
-> ip pim ssm group 225.0.0.0/24 priority 50
```

This command entry maps the multicast group 225.0.0.0/24 to PIM-SSM and specifies the priority value to be used for the entry as 50. This priority specifies the preference value to be used for this static configuration and provides fine control over which configuration is overridden by this static configuration. Values may range from 0 to 128. If the priority option has been defined a value of 65535 can be used to unset the priority.

You can also use the **override** parameter to specify whether or not this static configuration overrides the dynamically learned group mapping information for the specified group. As specifying the priority value obsoletes the **override** option, you can use only the **priority** parameter or the **override** parameter. By default, the **priority** option is not set and the **override** option is set to false.

Use the **no** form of this command to remove a static configuration of a SSM mode group mapping.

```
-> no ip pim ssm group 225.0.0.0/24
```

The default SSM address range (232.0.0.0 through 232.255.255.255) reserved by the Internet Assigned Numbers Authority is not enabled automatically for PIM-SSM and must be configured manually to support SSM. You can also map additional multicast address ranges for the SSM group. However, the multicast groups in the reserved address range can be mapped only to the SSM mode.

Verifying Group Mapping

To view PIM-DM group mappings, use the **show ip pim dense group** command. For example:

```
-> show ip pim dense group

Group Address/Pref Length   Mode  Override Precedence Status
---------------------------+-----+--------+----------+--------
225.0.0.0/24                  dm    false    none       enabled
```

To view PIM-SSM mode group mappings, use the **show ip pim ssm group** command. For example:

```
-> show ip pim ssm group

Group Address/Pref Length   Mode  Override Precedence Status
---------------------------+-----+--------+----------+--------
225.0.0.0/24                  ssm   false    none       enabled
```

Automatic Loading and Enabling of PIM after a System Reboot

If *any* PIM command is saved to the **boot.cfg** file in the post-boot running directory, the switch will automatically load PIM into memory. The post-boot running directory is the directory the switch will use as its running directory after the next switch reboot (i.e., Working or Certified).

If the command syntax **ip pim sparse admin-state enable** or **ip pim dense admin-state enable** is saved to the **boot.cfg** file in the post-boot running directory, the switch will automatically load PIM into memory *and* globally enable PIM the next time the switch reboots. For detailed information on the Working and Certified directories and how they are used, see the “CMM Directory Management” chapter in the *OmniSwitch AOS Release 8 Switch Management Guide*.

PIM Bootstrap and RP Discovery

Before configuring PIM-SM parameters, please consider the following important guidelines.

For correct operation, every PIM-SM router within a PIM-SM domain must be able to map a particular multicast group address to the same Rendezvous Point (RP). Otherwise, some receivers in the domain will not receive some groups. Two mechanisms are supported for multicast group address mapping:

- Bootstrap Router (BSR) Mechanism
- Static RP Configuration

The chosen multicast group address mapping mechanism should be used consistently throughout PIM-SM domain. Any RP address configured or learned *must* be a domain-wide reachable address.

Configuring a C-RP

Note. If you attempt to configure an interface that is not PIM enabled as a C-RP, you will receive the following error message:

```
ERROR: PIM is not enabled on this Interface
```

For information on enabling PIM on an interface, refer to page 7-20.

To configure the local router as the Candidate-Rendezvous Point (C-RP) for a specified IP multicast group(s), use the **ip pim candidate-rp** command. For example:

```
-> ip pim candidate-rp 50.1.1.1 225.16.1.1/32 priority 100 interval 100
```

This configures the switch to advertise the address 50.1.1.1 as the C-RP for the multicast group 225.16.1.1 with a mask of 255.255.255.255, set the priority level for this entry to 100, and set the interval at which the C-RP advertisements are sent to the Bootstrap Router to 100.

Use the **no** form of this command to remove the association of the device as a C-RP for a particular multicast group.

```
-> no ip pim candidate-rp 50.1.1.1 225.16.1.1/32
```

If no C-RP address is defined, the switch will not advertise itself as a C-RP for any groups. Only one RP address is supported per switch. If multiple candidate-RP entries are defined, they must use the same RP address.

The C-RP priority is used by a Designated Router to determine the RP for a particular group. As per the C-RP priority range, the lower the numerical value, the higher the priority. If two or more C-RPs have the same priority value and the same hash value, the C-RP with the highest IP address is selected by the DR.

Verifying C-RP Configuration

Check the C-RP address, priority level, and explicit multicast group information using the **show ip pim candidate-rp** command, as follows:

```
-> show ip pim candidate-rp
RP Address          Group Address   Priority  Interval  Status
------------------+---------------+---------+---------+--------
172.21.63.11        225.0.0.0/24      192        60         enabled
```

The group address is listed as 225.0.0.0. The class D group mask (255.255.255.255) has been translated into the Classless Inter-Domain Routing (CIDR) prefix length of /4. The C-RP is listed as 172.21.63.11. The status is enabled.

Specifying the Maximum Number of RPs

You can specify the maximum number of RPs allowed in a PIM-SM domain.

Important. PIM must be globally disabled on the switch before changing the maximum number of RPs. To disable PIM, use the **ip pim sparse admin-state** command. See "Disabling PIM Mode on the Switch" on page 7-21 for more information.

To specify a maximum number of RPs, use the **ip pim max-rps** command. For example:

```
-> ip pim max-rps 12
```

Note. This command is used with both IPv4 and IPv6 PIM-SM. PIM-SM must be disabled before changing **max-rps** value.

Verifying Maximum-RP Configuration

Check the maximum number of RPs using the **show ip pim sparse** command. For example:

```
-> show ip pim sparse
Status                     = enabled,
Keepalive Period           = 210,
Max RPs                    = 32,
Probe Time                 = 5,
Register Checksum          = header,
Register Suppress Timeout  = 60,
RP Threshold               = 1,
SPT Status                  = enabled,
```

For more information about these displays, see the "PIM Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Candidate Bootstrap Routers (C-BSRs)

A Candidate Bootstrap Router (C-BSR) is a PIM-SM-enabled router that is eligible for Bootstrap Router (BSR) status. To become a BSR, a C-BSR must be *elected*. A C-BSR sends Bootstrap messages to all neighboring routers. The messages include its IP address—which is used as an identifier—and its priority level. The C-BSR with the highest priority level is elected as the BSR by its neighboring routers. If there are multiple C-BSRs with the same highest priority, the C-BSR with the highest IP address will become the BSR.

For information on configuring a C-BSR, refer to "Configuring a C-BSR" below.

Configuring a C-BSR

You can use the **ip pim cbsr** command to configure the local router as the candidate-BSR for PIM domain. For example:

```
-> ip pim cbsr 50.1.1.1 priority 100 mask-length 4
```

This command specifies the router to use its local address 50.1.1.1 for advertising it as the candidate-BSR for that domain, the priority value of the local router as a C-BSR to be 100, and the mask-length that is advertised in the bootstrap messages as 4. The value of the priority is considered for the selection of C-BSR for PIM domain. The higher the value, the higher the priority.

Use the **no** form of this command to remove the local routers' candidacy as the BSR. For example:

```
-> no ip pim cbsr 50.1.1.1
```

Verifying the C-BSR Configuration

Check the C-BSR and information about priority and mask-length using the **show ip pim cbsr** command as follows:

```
-> show ip pim cbsr
CBSR Address              = 214.0.0.7,
Status                    = enabled,
CBSR Priority             = 0,
Hash Mask Length          = 30,
Elected BSR               = False,
Timer                     = 00h:00m:00s
```

For more information about these displays, see the "PIM Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Bootstrap Routers (BSRs)

As described in the "PIM Overview" section, the role of a Bootstrap Router (BSR) is to keep routers in the network "up to date" on reachable Candidate Rendezvous Points (C-RPs). BSRs are elected from a set of Candidate Bootstrap Routers (C-BSRs). Refer to page 7-9 for more information on C-BSRs.

Reminder. For correct operation, all PIM-SM routers within a PIM-SM domain must be able to map a particular multicast group address to the same Rendezvous Point (RP). PIM-SM provides two methods for group-to-RP mapping. One method is the Bootstrap Router mechanism, which also involves C-RP advertisements, as described in this section; the other method is static RP configuration.

A C-RP periodically sends out messages, known as *C-RP advertisements*. When a BSR receives one of these advertisements, the associated C-RP is considered reachable (if a valid route to the network exists). The BSR then periodically sends an updated list of reachable C-RPs to all neighboring routers in the form of a *Bootstrap message*.

The list of reachable C-RPs is also referred to as an *RP set*. To view the current RP set, use the **show ip pim group-map** command. For example:

```
-> show ip pim group-map
Origin    Group Address/Pref Length   RP Address    Mode  Precedence
--------+---------------------------+-------------+-----+-----------
BSR       225.0.0.0/24                172.21.63.11  asm   192
BSR       225.0.0.0/24                214.0.0.7     asm   192
Static    232.0.0.0/8                               ssm
```

For more information about these displays, see the "PIM Commands" chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Note. There is only one BSR per PIM-SM domain. This allows all PIM-SM routers in PIM-SM domain to view the same list of reachable C-RPs.

Configuring Static RP Groups

A static RP group is used in the group-to-RP mapping algorithm. To specify a static RP group, use the **ip pim static-rp** command. Be sure to enter a multicast group address, a corresponding group mask, and a 32-bit IP address for the static RP in the command line. For example:

```
-> ip pim static-rp 225.0.0.0/24 10.1.1.1 priority 10
```

This command entry maps all multicast groups 225.0.0.0/24 to the static RP 10.1.1.1 and specifies the priority value to be used for the static RP configuration as 10. This priority value provides fine control over which configuration is overridden by this static configuration. If the priority option has been defined, a value of 65535 can be used to unset the priority.

You can also specify this static RP configuration to override the dynamically learned RP information for the specified group using the **override** parameter. As specifying the priority value obsoletes the **override** option, you can use either the **priority** or **override** parameter only.

Use the **no** form of this command to delete a static RP configuration.

```
-> no ip pim static-rp 225.0.0.0/24 10.1.1.1
```

Verifying Static-RP Configuration

To view current Static RP Configuration settings, use the **show ip pim static-rp** command. For example:

```
-> show ip pim static-rp

Group Address/Pref Length   RP Address    Mode  Override Precedence Status
--------------------------+-------------+-----+--------+----------+--------
225.0.0.0/24                172.21.63.11  asm   false    none       enabled
```

Configuring PIM Anycast RP

Anycast RP provides load sharing and redundancy in Protocol Independent Multicast Sparse Mode (PIM-SM) networks. Anycast-RP is a mechanism that ISP-based backbones use to get fast convergence when a PIM rendezvous point (RP) router fails.

Anycast RP introduces the concept where the same IP address (RP Address) is configured on two or more routers serving as the RP. This address is advertised by the IGP. Other routers will select any of these routers based on the best path to the RP address. In case of a failure, the convergence is the same as the IGP. Anycast will be supported with PIM-SM only.

To configure the anycast RP set, which is the set of all routers that would act as the RP, use the **ip pim anycast-rp** command. For example:

```
-> ip pim anycast-rp 10.10.10.1 10.1.1.2
```

Use the **no** form of this command to delete an anycast RP configuration.

```
-> no ip pim anycast-rp 10.10.10.1 10.1.1.2
```

Verifying Anycast RP Configuration

To view current anycast RP configuration, use the **show ip pim anycast-rp** command. For example:

```
-> show ip pim anycast-rp
Anycast RP Address    Router Address      Local   Status
--------------------+-------------------+-------+---------
10.10.10.1           10.1.1.2            true    enabled
10.10.10.1           10.1.2.2            false   enabled
```

Note: The 'show ip pim anycast-rp' output will only exist on those routers participating in the Anycast-RP. All the PIM routers that are not participating in the Anycast-RP set will still have the PIM configuration defining the RP, but will not have the anycast-rp specific configuration.

PIM Anycast RP Configuration Example

Consider the following configuration example to configure Anycast-RP on two Routers, RP-Router-1 and RP-Router-2. Both routers will be used as the RP. The following configurations are required to configure Anycast-RP.

1 Configure RP address as 10.10.10.1, which will be configured on a Loopback1 interface on both the routers.

```
RP-Router-1-> ip interface "Loopback1" address 10.10.10.1
RP-Router-2-> ip interface "Loopback1" address 10.10.10.1
```

Configure OSPF on both routers so this Loopback1 address will then be advertised in OSPF to all routers in the network. Different PIM routers in the network will either reach RP-Router-1 or RP-Router-2 for the RP depending on the best path metric.

The Router ID used by the unicast routing protocols must not be the same as the IP address being used for this Anycast-RP address. The Loopback0 interface is often used as the Router ID by default if it is not explicitly configured on the system so it is recommended to use one of the additional LoopbackX interfaces for the Anycast-RP address to eliminate the potential for problems.

2 Configure the Anycast-RP address 10.10.10.1 on all routers using static RP configuration. The 224.0.0.0/4 specifies the group address range that the Anycast-RPs will be responsible for.

```
RP-Router-1-> ip pim static-rp 224.0.0.0/4 10.10.10.1
RP-Router-2-> ip pim static-rp 224.0.0.0/4 10.10.10.1
```

Note: This static configuration must exist on all PIM routers in the PIM domain, not just those routers that are participating in the Anycast-RP set.

3 Before defining the RP set, configure an additional LoopbackX interface on each of the routers, which is different than the Loopback1 interface that is being used as the RP address. In this example, Loopback0 is used. This Loopback0 address defines the IP address of the prospective RP and is used in communication between the different RPs in the Anycast-RP set. (Note: This Loopback0 address may be used as the Router ID for OSPF if it is not explicitly configured on the system.)

```
RP-Router-1-> ip interface "Loopback0" address 192.168.1.1
RP-Router-2-> ip interface "Loopback0" address 192.168.2.2
```

4 This Loopback0 address is used to complete the configuration of the RP set. Configure the anycast RP set, which is the set of all routers that would act as the RP. It is required to define your own IP address as well as all remote IP addresses in this RP set so the configuration for the Anycast-RP set will be the same on all RPs in the Anycast-RP set.

```
RP-Router-1-> ip pim anycast-rp 10.10.10.1 192.168.1.1
RP-Router-1-> ip pim anycast-rp 10.10.10.1 192.168.2.2

RP-Router-2-> ip pim anycast-rp 10.10.10.1 192.168.1.1
RP-Router-2-> ip pim anycast-rp 10.10.10.1 192.168.2.2
```

5 Following show output displays that the Anycast-RP set is the same on all routers.

```
RP-Router-1-> show ip pim anycast-rp
Anycast RP Address      Router Address      Local    Status
--------------------+--------------------+--------+----------
10.10.10.1            192.168.1.1          true     enabled
10.10.10.1            192.168.1.2          false    enabled
```

```
RP-Router-2-> show  ip pim anycast-rp
Anycast RP Address      Router Address      Local    Status
--------------------+--------------------+--------+----------
10.10.10.1            192.168.1.1          false    enabled
10.10.10.1            192.168.1.2          true     enabled
```

Group-to-RP Mapping

Using one of the mechanisms described in the sections above, a PIM-SM router receives one or more possible group-range-to-RP mappings. Each mapping specifies a range of multicast groups (expressed as a group and mask), as well as the RP to which such groups should be mapped. Each mapping may also have an associated priority. It is possible to receive multiple mappings—all of which might match the same multicast group. This is the common case with the BSR mechanism. The algorithm for performing the group-to-RP mapping is as follows:

1 Perform longest match on group-range to obtain a list of RPs.

2 From this list of matching RPs, find the one with the highest priority. Eliminate any RPs from the list that have lower priorities.

3 If only one RP remains in the list, use that RP.

4 If multiple RPs are in the list, use the PIM-SM hash function defined in the RFC to choose one. The RP with the highest resulting hash value is then chosen as the RP. If more than one RP has the same highest hash value, then the RP with the highest IP address is chosen.

This algorithm is invoked by a DR when it needs to determine an RP for a given group, such as when receiving a packet or an IGMP membership indication.

Configuring Keepalive Period

You can specify the duration for the Keepalive Timer using the ip pim keepalive-period command. This is the period during which the PIM router will maintain (S,G) state in the absence of explicit (S,G) local membership of (S,G) Join messages received to maintain it. For example,

```
-> ip pim keepalive-period 500
```

The above example configures the keepalive period as 500 seconds.

This timer is called the Keepalive period and Source Lifetime period in PIM-SM specification and PIM-DM specification, respectively.

Note. The value configured by the above command is common for PIM in the IPv4 as well as IPv6 environments.

Verifying Keepalive Period

To view the configured keepalive period, use the **show ip pim sparse** command. For example:

```
-> show ipv6 pim sparse

Status                      = enabled,
Keepalive Period            = 210,
Max RPs                     = 32,
Probe Time                  = 5,
Register Suppress Timeout   = 60,
RP Switchover               = enabled,
SPT Status                  = enabled,
```

You can also use the **show ip pim dense**, **show ipv6 pim sparse**, and **show ipv6 pim dense** commands to view the configured keepalive period.

Configuring Notification Period

The switch can be configured for a minimum time interval that must elapse between various notifications, such as neighbor loss notification, invalid register notification, invalid joinprune notification, RP mapping notification, and interface election notification. For example:

To set the time that must elapse between PIM neighbor loss notifications originated by the router, enter **ip pim neighbor-loss-notification-period** followed by the time in seconds. For example, to set the time period of 10 seconds, enter:

```
-> ip pim neighbor-loss-notification-period 10
```

To set the time that must elapse between PIM invalid register notifications originated by the router, enter **ip pim invalid-register-notification-period** followed by the time in seconds. For example, to set the time period of 100 seconds, enter:

```
-> ip pim invalid-register-notification-period 100
```

To set the time that must elapse between PIM invalid joinprune notifications originated by the router, enter **ip pim invalid-joinprune-notification-period** followed by the time.For example, to set the time period of 100 seconds, enter:

```
-> ip pim invalid-joinprune-notification-period 100
```

To set the time that must elapse between PIM RP mapping notifications originated by the router, enter **ip pim rp-mapping-notification-period** followed by the time in seconds.For example, to set the time period of 100 seconds, enter:

```
-> ip pim rp-mapping-notification-period 100
```

To set the time that must elapse between PIM interface election notifications originated by the router, enter **ip pim interface-election-notification-period** followed by the time in seconds. For example, to set the time period of 100 seconds, enter:

```
-> ip pim interface-election-notification-period 100
```

Note. The values configured by the above commands are common for PIM in the IPv4 as well as IPv6 environments.

Verifying the Notification Period

To view the configured notification period, use the **show ip pim notifications** command. For example:

```
-> show ip pim notifications

Neighbor Loss Notifications
  Period       = 0
  Count        = 0
Invalid Register Notifications
  Period       = 65535
  Msgs Rcvd    = 0
  Origin       = None
  Group        = None
  RP           = None
Invalid Join Prune Notifications
  Period       = 65535
  Msgs Rcvd    = 0
  Origin       = None
  Group        = None
  RP           = None
RP Mapping Notifications
  Period       = 65535
  Count        = 0
Interface Election Notifications
  Period       = 65535
  Count        = 0
```

PIM Multicast Scalability for Packed Register Messages

In PIM-SM networks, PIM Null Register messages are sent periodically from the first hop router to the RP to signal the presence of Multicast sources in the network and to keep the (S,G) state alive as long as the source is active.

Likewise, Register Stop messages are sent from the RP to stop the sending of the register encapsulated messages. These messages currently include information about a single multicast source and group. In large networks with a lot of sources, this can amount to a lot of PIM Control packets which ultimately may be dropped due to control plane processing overhead or CPU queue rate-limiting. The packing of these Null Registers and Register stops has been added to reduce the possibility of losing these packets.

Enabling PIM Register Packing

PIM Register Message Packing can be enabled or disabled using the **ip pim register-packing** command. For eg:

```
-> ip pim register-packing enable
-> ip pim register-packing force-enable
-> ip pim register-packing disable
```

When using this command with Anycast-RP, PIM register packing should be enabled only if it is supported by all PIM anycast RP members in the RP set for the RP address. When the value “force-enable” is used, the null registers will automatically be packed without waiting for acknowledgment from an RP on whether or not the packing of register messages is supported.It is recommended to use the configuration value of "force-enable' on all the routers in the domain, when register message packing is desired and Anycast-RP is used.

This command is supported only in the sparse mode.

Configuring the Maximum MTU size

To configure the maximum MTU size of packed PIM register messages sent out, use the command **ip pim register-mtu.** For eg:

```
-> ip pim register-mtu 1000
```

Before using this command, enable or force-enable PIM register packing by using the **ip pim register-packing** command. If the MTU size of the packet is larger than the interface that is used to send the packed messages, those packets may be dropped.

It is not recommended to configure to a large value unless it is known that all the RP routers in the domain can support the MTU size in order to avoid any fragmentation and reassembly of the packets.

This command is supported only in the sparse mode.

Configuring the Triggered Register Delay

To configure the Triggered Register Delay interval in milliseconds and to control the packing of initial bursts of triggered Null Register messages, use the command **ip pim register-delay**. For eg:

```
-> ip pim register-delay 100
```

This command is supported only in the sparse mode.

Verifying the PIM Packed Register Configuration

To view the configuration of PIM Register Packing and to verify the PIM Packed Register configuration, use the command **show ip pim sparse.** For eg:

```
-> show ip pim sparse
Status                      = disabled,
Keepalive Period            = 210,
Max RPs                     = 32,
Probe Time                  = 5,
Register Checksum           = header,
Register Suppress Timeout   = 60,
RP Threshold                = 1,
SPT Status                  = enabled,
BIDIR Status                = disabled,
BIDIR Periodic Interval     = 60,
BIDIR DF Abort Status       = disabled,
BFD Status                  = disabled,
MoFRR Status                = disabled,
MoFRR All Routes Status     = disabled,
MBR All Sources Status      = disabled,
MBR Operational Status      = disabled,
RP Hash Algorithm           = disabled,
ASM Fast Join               = disabled,
SSM Fast Join               = disabled,
BIDIR Fast Join             = disabled,
BIDIR SSM Compatibility     = enabled,
Register Rate Limit         = 100,
Join/Prune Message packing  = enabled,
Register Message packing    = enabled,
Register MTU                = 1500,
Register Triggered Delay    = 200
```

Enabling PIM Join/Prune Message Packing for IPv4

Whenever the Join/Prune messages are dropped due to control plane processing overhead or CPU queue rate-limiting, traffic may not converge until the next Join/Prune interval expires and the Join/Prune messages are resent. This can have negative affects when the primary unit of a virtual chassis goes down and the secondary must recover operation.

PIM message packing optimizes the time it takes to recover from these events by sending fewer packets, thus lowering the risk of buffer overflows from receiving too many packets to process when the network is trying to recover from a router failure.

To enable PIM Join/Prune message packing, use the **ip pim joinprune-packing** command. By default, this is enabled.

```
-> ip pim joinprune-packing enable
```

To disable PIM Join/Prune message packing, use the disable option in the **ip pim joinprune-packing** command. Disable action returns to the original operation of PIM where Join/Prune messages are not packed.

```
-> ip pim joinprune-packing disable
```

Note. This feature will only work with PIM-SM, PIM-SSM and PIM-BIDIR. This feature will not be supported with PIM-DM since there are no periodic Joins transmitted, but only explicitly triggered prunes and grafts.

Configure Join/Prune MTU

By default, the PIM Interface Join/Prune MTU value is '0' and the configured interface MTU value will be used in determining the maximum packet size that can be used in sending the packed Join/Prune messages. The PIM Interface Join/Prune MTU can be configured to help control the MTU size of the PIM Join/Prune packets getting sent out for cases where neighboring routers are not able to handle the large PIM Join/Prune packets efficiently. However, the actual maximum size used for PIM Join/Prune messages will be the smaller of the IP MTU value of the interface and the configured PIM interface Join/Prune MTU value.

To configure the PIM interface Join/Prune MTU value, use the **ip pim interface** command. For example,.

```
-> ip pim interface vlan-10 joinprune-mtu 1000
```

Configure Join/Prune Delay Interval

The Join/Prune delay interval is used to delay the sending of triggered Join/Prune messages and may be desirable to allow the packing of triggered Join/Prune messages due to bursts of protocol messages, which may result in subsequent bursts of triggered Join/Prune packets. The default value of ‘0’ implies no deferred processing and will result in no packing of triggered Join/Prune packets.

To configure the Join/Prune delay interval, use the **ip pim interface** command. For example,.

```
-> ip pim interface vlan-10 joinprune-delay 100
```

Verifying PIM Configuration

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

show ip pim sparse	Displays the status of the various global parameters for PIM-Sparse Mode.
show ip pim dense	Displays the status of the various global parameters for PIM-Dense Mode.
show ip pim ssm group	Displays the static configuration of multicast group mappings for PIM-Source-Specific Multicast (SSM) mode.
show ip pim dense group	Displays the static configuration of multicast group mappings for PIM-Dense Mode (DM).
show ip pim neighbor	Displays a list of active PIM neighbors.
show ip pim candidate-rp	Displays the IP multicast groups for which the local router advertises itself as a Candidate-RP.
show ip pim group-map	Displays the PIM group mapping table.
show ip pim interface	Displays detailed PIM settings for a specific interface. In general, it displays PIM settings for all the interfaces if no argument is specified.
show ip pim groute	Displays all (*,G) states that the IPv4 PIM has.
show ip pim sgroute	Displays all (S,G) states that the IPv4 PIM has.
show ip pim notifications	Displays the configuration of the configured notification periods as well as information on the events triggering the notifications.
show ip mroute	Displays multicast routing information for IP datagrams sent by particular sources to the IP multicast groups known to this router.
show ip pim static-rp	Displays the PIM Static RP table, which includes group address/mask, the static Rendezvous Point (RP) address, and the current status of Static RP configuration (i.e., enabled or disabled).
show ip pim bsr	Displays information about the elected BSR.
show ip pim cbsr	Displays the Candidate-BSR information that is used in the Bootstrap messages.

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

PIM for IPv6 Overview

IP version 6 (IPv6) is a new version of the Internet Protocol, designed as the successor to IP version 4 (IPv4), to overcome certain limitations in IPv4. IPv6 adds significant extra features that were not possible with IPv4. These include automatic configuration of hosts, extensive multicasting capabilities, and built-in security using authentication headers and encryption. Built-in support for QOS and path control are also features found in IPv6.

IPv6 is a hierarchical 128-bit addressing scheme that consists of 8 fields, composed of 16 bits each. An IPv6 address is written as a hexadecimal value (0-F) in groups of four, separated by colons. IPv6 provides 3x10^38 addresses, which can help overcome the shortage of IP addresses needed for Internet usage.

There are three types of IPv6 addresses: Unicast, Anycast, and Multicast. A Unicast address identifies a single interface, and a packet destined for a Unicast address is delivered to the interface identified by that address. An Anycast address identifies a set of interfaces, and a packet destined for an Anycast address is delivered to the nearest interface identified by that Anycast address. A Multicast address identifies a set of interfaces, and a packet destined for a Multicast address is delivered to all the interfaces identified by that Multicast address. There are no broadcast addresses in IPv6.

The current release also provides support for PIM to be configured in IPv6 environments using IPv6 multicast addresses. In the IPv6 addressing scheme, multicast addresses begin with the prefix ff00::/8. Similar to IPv6 unicast addresses, IPv6 multicast addresses also have different scopes depending on their prefix, though the range of possible scopes is different.

Multicast Listener Discovery (MLD) is the protocol used by an IPv6 router to discover the nodes that request multicast packets on its directly attached links and the multicast addresses that are of interest to those neighboring nodes. MLD is derived from version 2 of IPv4's Internet Group Management Protocol, IGMPv2. MLD uses ICMPv6 message types, rather than IGMP message types.

IPv6 PIM-SSM Support

IPv6 Protocol-Independent Multicast Source-Specific Multicast (IPv6 PIM-SSM) is a highly efficient extension of IPv6 PIM. SSM, using an explicit channel subscription model, allows receivers to receive multicast traffic directly from the source; an RP tree model is not used. In other words, a Shortest Path Tree (SPT) between the receiver and the source is created without the use of a Rendezvous Point (RP).

IPv6 PIM software supports Source-Specific Multicast. IPv6 PIM-SSM is automatically enabled and operational as long as IPv6 PIM is loaded (see page 7-6) and PIM sparse mode is enabled with the ssm group range defined (ip pim ssm group) and IGMPv3 source-specific joins are received within the SSM address range.

Source-Specific Multicast Addresses

The multicast addresses range FF3x::/32 that has been reserved by the Internet Assigned Numbers Authority (IANA) as Source-Specific Multicast (SSM) destination addresses is not enabled automatically and must be configured manually to support SSM. Addresses within this range are reserved for use by source-specific applications and protocols (e.g., IPv6 PIM-SSM) and cannot be used for any other functions or protocols. However, you can also map additional multicast address ranges for the SSM group.

Quick Steps for Configuring IPv6 PIM-DM

Note. PIM requires that IP Multicast Switching (IPMS) is enabled. IPMS is automatically enabled when a multicast routing protocol (either PIM or DVMRP) is enabled globally and on an interface *and* when the operational status of the interface is *up*. However, if you wish to manually enable IPMS on the switch, use the **ip multicast admin-state** command.

1 Manually load PIM into memory by entering the following command:

```
-> ip load pim
```

2 Create an IPv6 router interface on an existing VLAN by specifying a valid IPv6 address. To do this, use the **ipv6 interface** command. For example:

```
-> ipv6 interface vlan 1
-> ipv6 address 4132:86::19A/64 vlan 1
```

3 Enable PIM on the IPv6 interface using the **ipv6 pim interface** command. For example:

```
-> ipv6 pim interface vlan-1
```

Note. The IPv6 interface on which the PIM is enabled must already exist in the switch configuration.

4 Map the IPv6 PIM-Dense Mode (DM) protocol for a multicast group via the **ipv6 pim dense group** command. For example:

```
-> ipv6 pim dense group ff0e::1234/128
```

5 Globally enable the IPv6 PIM protocol by entering the following command.

```
-> ipv6 pim dense admin-state enable
```

6 Save your changes to the Working directory's **boot.cfg** file by entering the following command:

```
-> write memory
```

Note. *Optional*. To verify IPv6 PIM interface status, enter the **show ipv6 pim interface** command. The display is similar to the one shown below:

```
-> show ipv6 pim interface
Interface Name       Designated                 Hello    Join/Prune Oper
                     Router                     Interval Interval   Status
--------------------+--------------------------+--------+----------+---------
vlan-5               fe80::2d0:95ff:feac:a537   30       60         enabled
vlan-30              fe80::2d0:95ff:feac:a537   30       60         disabled
vlan-40              fe80::2d0:95ff:fee2:6eec   30       60         enabled
```

To verify global IPv6 PIM status, enter the **show ipv6 pim sparse** or **show ipv6 pim dense** command. The display for sparse mode is similar to the one shown below:

```
-> show ipv6 pim sparse
Status                      = enabled,
Keepalive Period            = 210,
Max RPs                     = 32,
Probe Time                  = 5,
Register Suppress Timeout   = 60,
RP Switchover               = enabled,
SPT Status                  = enabled,
```

The display for dense mode is similar to the one shown here:

```
-> show IPv6 pim dense
Status                       = enabled,
Source Lifetime              = 210,
State Refresh Interval       = 60,
State Refresh Limit Interval = 0,
State Refresh TTL            = 16
```

(additional table output not shown)

For more information about these displays, see the “PIM Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring IPv6 PIM

This section describes using the Command Line Interface (CLI) commands to complete the following steps to configure PIM in an IPv6 environment:

- Enabling/disabling IPv6 PIM on a specific interface
- Enabling/disabling IPv6 PIM mode on the switch
- IPv6 PIM Bootstrap and RP Discovery
- Configuring a C-RP for IPv6 PIM
- Configuring Candidate Bootstrap Routers (C-BSRs) for IPv6 PIM
- Configuring static RP groups for IPv6 PIM
- Configuring RP-switchover for IPv6 PIM

Enabling IPv6 PIM on a Specific Interface

IPv6 PIM must be enabled on an interface using the **ipv6 pim interface** command. An interface can be any IPv6 router interface that has been assigned to an existing VLAN. For information on assigning a router interface to a VLAN, refer to the "Configuring IPv6" chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide.*

To enable PIM on a specific IPv6 interface, use the **ipv6 pim interface** command. The interface identifier used in the command syntax is the valid interface name of an existing IPv6 VLAN router interface. For example:

```
-> ipv6 pim interface vlan-2
```

Disabling IPv6 PIM on a Specific Interface

To disable PIM on a specific IPv6 interface, use the **no ipv6 pim interface** command. Be sure to include the name of the interface. For example:

```
-> no ipv6 pim interface vlan-2
```

Viewing IPv6 PIM Status and Parameters for a Specific Interface

To view the current IPv6 PIM interface information—which includes IPv6 addresses for PIM-enabled interfaces, Hello and Join/Prune intervals, and current operational status—use the **show ipv6 pim interface** command. For example:

```
-> show ipv6 pim interface
Interface Name       Designated                Hello    Join/Prune Oper
                     Router                    Interval Interval   Status
-------------------+-------------------------+--------+----------+---------
vlan-5               fe80::2d0:95ff:feac:a537    30       60         enabled
vlan-30              fe80::2d0:95ff:feac:a537    30       60         disabled
vlan-40              fe80::2d0:95ff:fee2:6eec    30       60         enabled
```

Enabling IPv6 PIM Mode on the Switch

To globally enable IPv6 PIM-Sparse Mode on the switch, use the **ipv6 pim sparse admin-state** command. Enter the command syntax as shown below:

```
-> ipv6 pim sparse admin-state enable
```

To globally enable IPv6 PIM-Dense Mode on the switch, use the **ipv6 pim dense admin-state** command. Enter the command syntax as shown below:

```
-> ipv6 pim dense admin-state enable
```

Disabling IPv6 PIM Mode on the Switch

To globally disable IPv6 PIM-Sparse Mode on the switch, use the **ipv6 pim sparse admin-state** command. Enter the command syntax as shown below:

```
-> ipv6 pim sparse admin-state disable
```

To globally disable IPv6 PIM-Dense Mode on the switch, use the **ipv6 pim dense admin-state** command. Enter the command syntax as shown below:

```
-> ipv6 pim dense admin-state disable
```

Checking the Current Global IPv6 PIM Status

To view the current global IPv6 PIM status, as well as additional global IPv6 PIM settings, use the **show ip pim sparse** or **show ip pim dense** command. For example:

```
-> show ipv6 pim sparse
Status                      = enabled,
Keepalive Period            = 210,
Max RPs                     = 32,
Probe Time                  = 5,
Register Suppress Timeout   = 60,
RP Switchover               = enabled,
SPT Status                   = enabled,

-> show ipv6 pim dense
Status                        = enabled,
Source Lifetime               = 210,
State Refresh Interval        = 60,
State Refresh Limit Interval  = 0,
State Refresh TTL              = 16
```

Mapping an IPv6 Multicast Group to a PIM Mode

PIM mode is an attribute of the IPv6 multicast group mapping and cannot be configured on an interface basis. The Dense mode or Source-Specific Multicast mode can be configured only on an IPv6 multicast group basis.

Mapping an IPv6 Multicast Group to PIM-DM

To statically map an IPv6 multicast group(s) to PIM-Dense Mode (DM), you can use the **ipv6 pim dense group** command. For example:

```
-> ipv6 pim dense group ff0e::1234/128 priority 50
```

This command maps the multicast group `ff0e::1234/128` to PIM-DM and assigns a priority value of 50 to the entry. This priority specifies the preference value to be used for this static configuration and provides fine control over which configuration is overridden by this static configuration. Values may range from 0 to 128. If the priority option has been defined, a value of 65535 can be used to unset the priority

You can also use the **override** parameter to specify whether or not this static configuration overrides the dynamically learned group mapping information for the specified group. As specifying the priority value obsoletes the **override** option, you can use only the **priority** parameter or the **override** parameter. By default, the **priority** option is not set and the **override** option is set to false.

Use the **no** form of this command to remove a static configuration of a dense mode group mapping.

```
-> no ipv6 pim dense group ff0e::1234/128
```

Mapping an IPv6 Multicast Group to PIM-SSM

To statically map an IPv6 multicast group(s) to PIM-Source-Specific Multicast mode (SSM), you can use the **ipv6 pim ssm group** command. For example:

```
-> ipv6 pim ssm group ff30::1234:abcd/128 priority 50
```

This command entry maps the multicast group `ff30::1234:abcd/128` to PIM-SSM mode and specifies the priority value to be used for the entry as 50. This priority specifies the preference value to be used for this static configuration and provides fine control over which configuration is overridden by this static configuration. Values may range from 0 to 128. If the priority option has been defined, a value of 65535 can be used to un-set the priority.

You can also use the **override** parameter to specify whether or not this static configuration overrides the dynamically learned group mapping information for the specified group. As specifying the priority value obsoletes the **override** option, you can use only the **priority** parameter or the **override** parameter. By default, the **priority** option is not set and the **override** option is set to false.

Use the **no** form of this command to remove a static configuration of a SSM mode group mapping.

```
-> no ipv6 pim ssm group ff30::1234:abcd/128
```

The default SSM address range (FF3x::/32) reserved by the Internet Assigned Numbers Authority is not enabled automatically for PIM-SSM and must be configured manually to support SSM. You can also map additional IPv6 multicast address ranges for the SSM group using this command. However, the IPv6 multicast groups in the reserved address range can be mapped only to the SSM mode.

Verifying Group Mapping

To display the static configuration of IPv6 multicast group mappings for PIM-Dense Mode (DM), use the **show ipv6 pim dense group** command. For example:

```
-> show ipv6 pim dense group
Group Address/Pref Length   Mode  Override Precedence Status
--------------------------+-----+--------+----------+--------
ff00::/8                    dm    false    none       enabled
ff34::/32                   dm    false    none       enabled
```

To display the static configuration of IPv6 multicast group mappings for PIM-Source-Specific Multicast (SSM) mode, use the **show ipv6 pim ssm group** commad. For example:

```
-> show ipv6 pim ssm group
Group Address/Pref Length   Mode  Override Precedence Status
--------------------------+-----+--------+----------+--------
ff00::/8                    ssm    false    none       enabled
ff34::/32                   ssm    false    none       enabled
```

IPv6 PIM Bootstrap and RP Discovery

Before configuring IPv6 PIM-SM parameters, please consider the following important guidelines.

For correct operation, every IPv6 PIM-SM router within an IPv6 PIM-SM domain must be able to map a particular multicast group address to the same Rendezvous Point (RP). Otherwise, some receivers in the domain will not receive some groups. Two mechanisms are supported for multicast group address mapping:

- Bootstrap Router (BSR) Mechanism
- Static RP Configuration

The chosen multicast group address mapping mechanism should be used consistently throughout the IPv6 PIM-SM domain. Any RP address configured or learned *must* be a domain-wide reachable address.

Configuring a C-RP for IPv6 PIM

To configure the local router as the Candidate-Rendezvous Point (C-RP) for a specified IPv6 multicast group(s), use the **ipv6 pim candidate-rp** command. For example:

```
-> ipv6 pim candidate-rp 2000::1 ff0e::1234/128 priority 100 interval 100
```

This specifies the switch to advertise the address 2000::1 as the C-RP for the multicast group ff0e::1234 with a prefix length of 128, set the priority level for this entry to 100, and set the interval at which the C-RP advertisements are sent to the bootstrap router to 100.

Use the **no** form of this command to remove the association of the device as a C-RP for a particular multicast group.

```
-> no ipv6 pim candidate-rp 2000::1 ff0e::1234/128
```

If no C-RP address is defined, the switch will not advertise itself as a C-RP for any groups.

The C-RP priority is used by a Designated Router to determine the RP for a particular group. As per the C-RP priority range, the lower the numerical value, the higher the priority. If two or more C-RPs have the same priority value and the same hash value, the C-RP with the highest IPv6 address is selected by the DR.

There may be multiple C-RPs defined for IPv6 in order to support different C-RPs for different zones. A particular C-RP will unicast the C-RP-Adv messages to the BSR for each scope zone for which it has state.

Verifying the Changes

Check the maximum number of RPs using the **show ipv6 pim sparse** command. For example:

```
-> show ipv6 pim sparse
Status                      = enabled,
Keepalive Period            = 210,
Max RPs                     = 32,
Probe Time                  = 5,
Register Suppress Timeout   = 60,
RP Switchover               = enabled,
SPT Status                  = enabled,
```

Check C-RP address, priority level, and explicit multicast group information using the **show ipv6 pim candidate-rp** command:

```
-> show ipv6 pim candidate-rp
RP Address         Group Address   Priority  Interval  Status
-----------------+---------------+---------+---------+--------
3000::11           FF00::/8        192       60        enabled
```

The group address is listed as FF00::/8. The C-RP is listed as 3000::11. The status is enabled.

For more information about these displays, see the “PIM Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuring Candidate Bootstrap Routers (C-BSRs) for IPv6 PIM

You can use the **ipv6 pim cbsr** command to configure the local router as the candidate-BSR for the IPv6 PIM domain. For example:

```
-> ipv6 pim cbsr 2000::1 priority 100 mask-length 4
```

This command specifies the router to use its local address 2000::1 for advertising it as the C-BSR for that domain, sets the priority value of the local router as a C-BSR to 100, and sets the mask-length that is advertised in the bootstrap messages to 4. The priority value is used to select a C-BSR for the IPv6 PIM domain. The higher the value, the higher the priority.

Use the **no** form of this command to remove the local routers' candidacy as the BSR. For example:

```
-> no ipv6 pim cbsr 2000::1
```

Verifying the C-BSR Configuration

Check C-BSR and information about priority and mask-length using the **show ipv6 pim cbsr** command, as follows:

```
-> show ipv6 pim cbsr
CBSR Address              = 3000::7,
Status                    = enabled,
CBSR Priority             = 0,
Hash Mask Length          = 126,
Elected BSR               = False,
Timer                     = 00h:00m:00s
```

For more information about these displays, see the “PIM Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Bootstrap Routers (BSRs)

As described in the “PIM Overview” section, the role of a Bootstrap Router (BSR) is to keep routers in the network “up to date” on reachable Candidate Rendezvous Points (C-RPs). BSRs are elected from a set of Candidate Bootstrap Routers (C-BSRs). Refer to page 7-9 for more information on C-BSRs.

Reminder. For correct operation, all IPv6 PIM-SM routers within an IPv6 PIM-SM domain must be able to map a particular multicast group address to the same Rendezvous Point (RP). PIM-SM provides two methods for group-to-RP mapping. One method is the Bootstrap Router mechanism, which also involves C-RP advertisements, as described in this section; the other method is static RP configuration. Note that, if static RP configuration is enabled, the Bootstrap mechanism and C-RP advertisements *are automatically disabled*. For more information on static RP status and configuration, refer to “Configuring Static RP Groups” below.

A C-RP periodically sends out messages, known as *C-RP advertisements*. When a BSR receives one of these advertisements, the associated C-RP is considered reachable (if a valid route to the network exists). The BSR then periodically sends an updated list of reachable C-RPs to all neighboring routers in the form of a *Bootstrap message*.

Note. The list of reachable C-RPs is also referred to as an *RP set*. To view the current RP set, use the **show ipv6 pim group-map** command. For example:

```
-> show ipv6 pim group-map
Origin    Group Address/Pref Length   RP Address    Mode  Precedence
---------+--------------------------+-------------+-----+-----------
BSR       ff00::/8                    3000::11      asm   192
BSR       ff00::/8                    4000::7       asm   192
SSM       ff33::/32                                 ssm
```

For more information about these displays, see the “PIM Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Note. There is only one BSR per IPv6 PIM-SM domain within the default scope. This allows all IPv6 PIM-SM routers in the IPv6 PIM-SM domain to view the same list of reachable C-RPs.

Configuring Static RP Groups for IPv6 PIM

A static RP group is used in the group-to-RP mapping algorithm. To specify a static RP group, use the **ipv6 pim static-rp** command. Be sure to enter a multicast group address, a corresponding group mask, and a 128-bit IPv6 address for the static RP in the command line. For example:

```
-> ipv6 pim static-rp ff0e::1234/128 2000::1 priority 10
```

This command entry maps all multicast groups ff0e::1234/128 to the static RP 2000::1 and specifies the priority value to be used for the static RP configuration as 10. This priority value provides fine control over which configuration is overridden by this static configuration. If the priority option has been defined, a value of 65535 can be used to unset the priority

You can also specify whether or not this static RP configuration to override the dynamically learned RP information for the specified group using the **override** parameter. As specifying the priority value obsoletes the **override** option, you can use either the **priority** or **override** parameter only.

Use the **no** form of this command to delete a static RP configuration.

```
-> no ipv6 pim static-rp ff0e::1234/128 2000::1
```

To view current Static RP Configuration settings, use the **show ipv6 pim static-rp** command.

Configuring Anycast RP for IPv6 PIM

Anycast RP provides load sharing and redundancy in Protocol Independent Multicast Sparse Mode (PIM-SM) networks. Anycast-RP is a mechanism that ISP-based backbones use to get fast convergence when a PIM rendezvous point (RP) router fails.

Anycast RP introduces the concept where the same IPv6 address (RP Address) is configured on two or more routers serving as the RP. This address is advertised by the IGP. Other routers will select any of these routers based on the best path to the RP address. In case of a failure, the convergence is the same as the IGP. Anycast will be supported with PIM-SM only.

To configure anycast RP set, which is the set of all routers that would act as the RP, use the **ipv6 pim anycast-rp** command. For example:

```
-> ipv6 pim anycast-rp 2001:1::1 3001:1::1
```

Use the no form of this command to delete an anycast RP configuration.

```
-> no ipv6 pim anycast-rp 2001:1::1 3001:1::1
```

Verifying Anycast RP Configuration

To view current anycast RP configuration, use the **show ipv6 pim anycast-rp** command. For example:

```
-> show ipv6 pim anycast-rp
Anycast RP Address        Router Address   Local   Status
--------------------+------------------+-------+--------
2001:1::1                  3001:1::1       true    enabled
2001:1::1                  3001:1::2       false   enabled
```

Note: The 'show ipv6 pim anycast-rp' output will only exist on those routers participating in the Anycast-RP. All the PIM routers that are not participating in the Anycast-RP set will still have the PIM configuration defining the RP, but will not have the anycast-rp specific configuration.

Group-to-RP Mapping

Using one of the mechanisms described in the sections above, an IPv6 PIM-SM router receives one or more possible group-range-to-RP mappings. Each mapping specifies a range of IPv6 multicast groups (expressed as a group and mask), as well as the RP to which such groups should be mapped. Each mapping may also have an associated priority. It is possible to receive multiple mappings—all of which might match the same multicast group. This is the common case with the BSR mechanism. The algorithm for performing the group-to-RP mapping is as follows:

1 Perform longest match on group-range to obtain a list of RPs.

2 From this list of matching RPs, find the one with the highest priority. Eliminate any RPs from the list that have lower priorities.

3 If only one RP remains in the list, use that RP.

4 If multiple RPs are in the list, use the PIM-SM hash function defined in the RFC to choose one. The RP with the highest resulting hash value is then chosen as the RP. If more than one RP has the same highest hash value, then the RP with the highest IPv6 address is chosen.

This algorithm is invoked by a DR when it needs to determine an RP for a given group, such as when receiving a packet or an IGMP membership indication.

Configuring RP-Switchover for IPv6 PIM

You can configure an RP to attempt switching to native forwarding upon receiving the first register-encapsulated packet from the source DR in the IPv6 PIM domain. For example:

```
-> ipv6 pim rp-switchover enable
```

The above command enables the RP to switch to native forwarding.

```
-> ipv6 pim rp-switchover disable
```

The above command disables the RP from switching to native forwarding.

You cannot specify a pre-configured threshold, such as the RP threshold, as you would do for IPv4 PIM.

Verifying RP-Switchover

To view the status of the RP-switchover capability, use the **show ipv6 pim sparse** command.

```
-> show ipv6 pim sparse
Status                      = enabled,
Keepalive Period            = 210,
Max RPs                     = 32,
Probe Time                  = 5,
Register Suppress Timeout   = 60,
RP Switchover               = enabled,
SPT Status                   = enabled
```

IPv6 PIM Multicast Scalability for Packed Register Messages

In PIM-SM networks, PIM Null Register messages are sent periodically from the first hop router to the RP to signal the presence of Multicast sources in the network and to keep the (S,G) state alive as long as the source is active.

Likewise, Register Stop messages are sent from the RP to stop the sending of the register encapsulated messages. These messages currently include information about a single multicast source and group. In large networks with a lot of sources, this can amount to a lot of PIM Control packets which ultimately may be dropped due to control plane processing overhead or CPU queue rate-limiting. The packing of these Null Registers and Register stops has been added to reduce the possibility of losing these packets.

Enabling IPv6 PIM Register Packing

PIM Register Message Packing can be enabled or disabled using the **ipv6 pim register-packing** command. For eg:

```
-> ipv6 pim register-packing enable
-> ipv6 pim register-packing force-enable
-> ipv6 pim register-packing disable
```

When using this command with Anycast-RP, PIM register packing should be enabled only if it is supported by all PIM anycast RP members in the RP set for the RP address. When the value “force-enable” is used, the null registers will automatically be packed without waiting for acknowledgment from an RP on whether or not the packing of register messages is supported. It is recommended to use the configuration value of “force-enable” on all the routers in the domain, when register message packing is desired and Anycast-RP is used.

This command is supported only in the sparse mode.

Configuring the Maximum MTU size for PIM IPv6

To configure the maximum MTU size of packed PIM register messages sent out, use the command **ipv6 pim register-mtu.** For eg:

```
-> ipv6 pim register-mtu 1000
```

Before using this command, enable or force-enable PIM register packing by using the **ip pim register-packing** command. If the MTU size of the packet is larger than the interface that is used to send the packed messages, those packets may be dropped.

It is not recommended to configure to a large value unless it is known that all the RP routers in the domain can support the MTU size in order to avoid any fragmentation and reassembly of the packets.

This command is supported only in the sparse mode.

Configuring the Triggered Register Delay for PIM IPv6

To configure the Triggered Register Delay interval in milliseconds and to control the packing of initial bursts of triggered Null Register messages, use the command **ipv6 pim register-delay**. For eg:

```
-> ipv6 pim register-delay 100
```

This command is supported only in the sparse mode.

Verifying the PIM IPv6 Packed Register Configuration

To view the configuration of IPv6 PIM Register Packing and to verify the PIM IPv6 Packed Register configuration, use the command **show ipv6 pim sparse.** For eg:

```
-> show ipv6 pim sparse
Status                       = disabled,
Keepalive Period             = 210,
Max RPs                      = 32,
Probe Time                   = 5,
Register Suppress Timeout    = 60,
RP Switchover                = enabled,
SPT Status                   = enabled,
BIDIR Status                 = disabled,
BIDIR Periodic Interval      = 60,
BIDIR DF Abort Status        = disabled,
BFD Status                   = disabled,
ASM Fast Join                = disabled,
SSM Fast Join                = disabled,
BIDIR Fast Join              = disabled,
BIDIR SSM Compatibility      = disabled,
Register Rate Limit          = 100,
Join/Prune Message Packing   = enabled,
Register Message packing     = force-enabled,
Register MTU                 = 1500,
Register Triggered Delay     = 200
```

Enabling PIM Join/Prune Message Packing for IPv6

Whenever the Join/Prune messages are dropped due to control plane processing overhead or CPU queue rate-limiting, traffic may not converge until the next Join/Prune interval expires and the Join/Prune messages are resent. This can have negative affects when the primary unit of a virtual chassis goes down and the secondary must recover operation.

PIM message packing optimizes the time it takes to recover from these events by sending fewer packets, thus lowering the risk of buffer overflows from receiving too many packets to process when the network is trying to recover from a router failure.

To enable PIM Join/Prune message packing, use the **ipv6 pim joinprune-packing** command. By default, this is enabled.

```
-> ipv6 pim joinprune-packing enable
```

To disable PIM Join/Prune message packing, use the disable option in the **ipv6 pim joinprune-packing** command. Disable action returns to the original operation of PIM where Join/Prune messages are not packed.

```
-> ipv6 pim joinprune-packing disable
```

Note. This feature will only work with PIM-SM, PIM-SSM and PIM-BIDIR. This feature will not be supported with PIM-DM since there are no periodic Joins transmitted, but only explicitly triggered prunes and grafts.

Configure Join/Prune MTU

By default, the PIM Interface Join/Prune MTU value is '0' and the configured interface MTU value will be used in determining the maximum packet size that can be used in sending the packed Join/Prune messages. The PIM Interface Join/Prune MTU can be configured to help control the MTU size of the PIM Join/Prune packets getting sent out for cases where neighboring routers are not able to handle the large PIM Join/Prune packets efficiently. However, the actual maximum size used for PIM Join/Prune messages will be the smaller of the IP MTU value of the interface and the configured PIM interface Join/Prune MTU value.

To configure the PIM interface Join/Prune MTU value, use the **ipv6 pim interface** command. For example,.

```
-> ipv6 pim interface vlan-10 joinprune-mtu 1000
```

Configure Join/Prune Delay Interval

The Join/Prune delay interval is used to delay the sending of triggered Join/Prune messages and may be desirable to allow the packing of triggered Join/Prune messages due to bursts of protocol messages, which may result in subsequent bursts of triggered Join/Prune packets. The default value of ‘0’ implies no deferred processing and will result in no packing of triggered Join/Prune packets.

To configure the Join/Prune delay interval, use the **ipv6 pim interface** command. For example,.

```
-> ipv6 pim interface vlan-10 joinprune-delay 100
```

Verifying IPv6 PIM Configuration

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

show ipv6 pim sparse	Displays the status of the various global parameters for the IPv6 PIM-Sparse Mode.
show ipv6 pim dense	Displays the status of the various global parameters for the IPv6 PIM-Dense Mode.
show ipv6 pim ssm group	Displays the static configuration of IPv6 multicast group mappings for PIM-Source-Specific Multicast (SSM).
show ipv6 pim dense group	Displays the static configuration of IPv6 multicast group mappings for PIM-Dense Mode (DM).
show ipv6 pim neighbor	Displays a list of active IPv6 PIM neighbors.
show ipv6 pim candidate-rp	Displays the IPv6 multicast groups for which the local router advertises itself as a Candidate-RP.
show ipv6 pim group-map	Displays the IPv6 PIM group mapping table.
show ipv6 pim interface	Displays detailed IPv6 PIM settings for a specific interface. In general, it displays IPv6 PIM settings for all the interfaces if no argument is specified.
show ipv6 pim groute	Displays all (*,G) states that IPv6 PIM has.
show ipv6 pim sgroute	Displays all (S,G) states that IPv6 PIM has.
show ip pim notifications	Displays the configuration of the configured notification periods as well as information on the events triggering the notifications.
show ipv6 mroute	Displays multicast routing information for IPv6 datagrams sent by particular sources to the IPv6 multicast groups known to this router.
show ipv6 pim static-rp	Displays the IPv6 PIM Static RP table, which includes IPv6 multicast group address/prefix length, the static Rendezvous Point (RP) address, and the current status of the static RP configuration (i.e., enabled or disabled).
show ipv6 pim bsr	Displays information about the elected IPv6 BSR.
show ipv6 pim cbsr	Displays the IPv6 Candidate-BSR information that is used in the Bootstrap messages.

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

8 Configuring a Multicast Border Router

The OmniSwitch support of interoperability between Protocol-Independent Multicast (PIM) and Distance Vector Multicast Routing Protocol (DVMRP) is based on rules defined in RFC 2715 and multicast border router (MBR) functionality defined in the PIM-SM specification (RFC 4601). The supported MBR functionality allows receivers and sources within PIM and DVMRP domains to communicate and satisfy RFC 2715 rules.

In This Chapter

This chapter provides information about how to configure the OmniSwitch as a Multicast Border Router to provide interoperability between PIM and DVMRP domains. CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- “Quick Steps for Configuring an OmniSwitch MBR” on page 8-2.
- “Multicast Border Router Overview” on page 8-3.
- “Configuring a Multicast Border Router” on page 8-4.
- “Verifying the MBR Configuration” on page 8-8.

For more information about PIM, see Chapter 7, “Configuring PIM.”

For more information about DVMRP, see Chapter 6, “Configuring DVMRP.”

Quick Steps for Configuring an OmniSwitch MBR

1 Before attempting to configure multicast border router functionality, be sure to manually load and configure both PIM and DVMRP on the switch. For example:

```
-> ip load pim
-> ip pim interface vlan-2
-> ip pim interface vlan-3
-> ip pim dense group 225.0.0.0/24
-> ip pim dense admin-state enable

-> ip load dvmrp
-> ip dvmrp interface vlan-4
-> ip dvmrp interface vlan-6
-> ip dvmrp admin-state enable
-> ip mroute mbr admin-state enable
```

2 Configure the switch to operate as a multicast border router using the **ip mroute mbr** command. For example:

```
-> ip mroute mbr admin-state enable
```

Note. *Optional*. To verify the multicast border router status, enter the **show ip mroute mbr** command. The display is similar to the one shown here:

```
-> show ip mroute mbr
MBR Status                      = enabled,
Protocols Registered            = DVMRP PIM
```

MBR functionality is operationally active when the feature is administratively enabled and both PIM and DVMRP are listed as registered protocols.

For more information about this display, see the “Multicast Routing Commands” chapter in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

Multicast Border Router Overview

The multicast border router (MBR) functionality implemented for the OmniSwitch supports interoperability between a PIM and DVMRP domain. Interoperability between PIM and other protocols or between multiple PIM domains is not supported. In addition, PIM support refers only to PIM-DM and PIM-SM (PIM-SSM is not supported).

Based on the RFC 2715 definition of multicast border router, an OmniSwitch MBR consists of a DVMRP instance and a PIM instance with one or more active interfaces in each instance. In this role, the OmniSwitch MBR performs the following primary functions:

- The MBR first pulls down packets generated within the PIM domain and injects them into the DVMRP domain.
- The MBR then imports packets generated within the DVMRP domain so that they can be delivered to group members inside the PIM domain, using PIM mechanisms.
- In the case of transit networks, the MBR passes the multicast traffic through the PIM and DVMRP domains.

Multicast border router functionality is configured and enabled on an OmniSwitch that is located at points where PIM and DVMRP regions interconnect. When MBR is operationally active, sources and receivers from a DVMRP domain can communicate with sources and receivers inside a PIM domain.

DVMRP Overview

This section provides a brief overview of the OmniSwitch DVMRP implementation. For more detailed information about using DVMRP, see Chapter 6, “Configuring DVMRP.”

Distance Vector Multicast Routing Protocol (DVMRP) Version 3 is a multicast routing protocol that enables routers to efficiently propagate IP multicast traffic through a network. Multicast traffic consists of a data stream that originates from a single source and is sent to hosts that have subscribed to that stream. Live video broadcasts, video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news services are examples of multicast traffic. Multicast traffic is distinguished from unicast traffic and broadcast traffic as follows:

- Unicast traffic is addressed to a single host.
- Broadcast traffic is transmitted to all hosts.
- Multicast traffic is transmitted to a subset of hosts (the hosts that have subscribed to the multicast data stream).

DVMRP is a distributed multicast routing protocol that dynamically generates per-source delivery trees based upon routing exchanges, using a technique called *Reverse Path Multicasting*. When a multicast source begins to transmit, the multicast data is flooded down the delivery tree to all points in the network. DVMRP then *prunes* (i.e., removes branches from) the delivery tree where the traffic is unwanted.

Pruning continues to occur as group membership changes or routers determine that no group members are present. This restricts the delivery trees to the minimum branches necessary to reach all group members, thus optimizing router performance. New branches can also be added to the delivery trees dynamically as new members join the multicast group. The addition of new branches is referred to as *grafting*.

PIM Overview

This section provides a brief overview of the OmniSwitch PIM implementation. The use of PIM throughout this chapter refers to PIM-DM and PIM-SM. For more detailed information about using PIM-SM and PIM-DM, see Chapter 7, "Configuring PIM."

Protocol-Independent Multicast (PIM) is an IP multicast routing protocol that uses routing information provided by unicast routing protocols, such as RIP and OSPF. Note that PIM is not dependent on any particular unicast routing protocol. Sparse mode PIM (PIM-SM) contrasts with flood-and-prune dense mode multicast protocols, such as DVMRP and PIM Dense Mode (PIM-DM), in that multicast forwarding in PIM-SM is initiated only through specific requests.

Downstream routers must explicitly join PIM-SM distribution trees to receive multicast streams on behalf of directly connected receivers or other downstream PIM-SM routers. This paradigm of receiver-initiated forwarding makes PIM ideal for network environments where receiver groups are thinly populated and bandwidth conservation is a concern, such as in wide area networks (WANs). PIM-DM uses RPF (Reverse Path Forwarding) to prevent looping of multicast datagrams while flooding. If some areas of the network do not have group members, PIM-DM will prune the forwarding branch by instantiating the prune state.

PIM-DM differs from PIM-SM in two essential ways:

- There are no periodic joins transmitted, only explicitly triggered prunes and grafts.
- There is no Rendezvous Point (RP). This is particularly important in networks that cannot tolerate a single point of failure.

Configuring a Multicast Border Router

Before configuring this feature, PIM and DVMRP must first be loaded to memory via the **ip load** command. In addition, an active PIM and DVMRP interface is required for these protocols to register with the multicast border router (MBR) functionality. PIM and DVMRP have to register with the MBR so that MBR can provide the functionality needed to support interoperability between the two protocols.

PIM and DVMRP are dynamically registered with MBR as soon as the first interface is enabled and operational for the particular protocol. In other words, as soon as the output of **show ip mroute interface** command has the first interface enabled for the protocol, then the output of **show ip mroute mbr** will show the protocol registered. For example:

```
-> show ip mroute interface

Interface Name     IP Address     TTL   Multicast Protocol
---------------+---------------+------+--------------------
vlan-4             214.0.0.7       0        PIM
vlan-6             172.21.63.7     0       DVMRP

-> show ip mroute mbr
MBR Status                      = enabled,
Protocols Registered            = DVMRP PIM
```

Note. It is possible for MBR to be enabled, but until both PIM and DVMRP have enabled at least one interface and are active, then MBR functionality is still not operational.

Enabling/Disabling MBR

By default, MBR is disabled for the switch. To enable MBR functionality, use the **ip mroute mbr** command. For example:

```
-> ip mroute mbr admin-state enable
```

To disable MBR functionality for the switch, use the **ip mroute mbr** command with the **disable** option. For example:

```
-> ip mroute mbr admin-state enable
```

Configuring PIM Route Notification

When MBR functionality is operationally active for the switch (PIM and DVMRP active and registered) PIM notifies DVMRP about the routes for subnets directly connected to PIM interfaces. To configure PIM to notify DVMRP about the routes to all multicast sources learned, use the **ip pim mbr all-sources** command. For example:

```
-> ip pim mbr all-sources
```

To revert route notification back to the default, use the **no** form of the **ip pim mbr all-sources** command. For example:

```
-> no ip pim mbr all-sources
```

DVMRP advertises the routes received from PIM within the DVMRP domain using standard DVMRP mechanisms.

Use the **show ip pim sparse** or the **show ip pim dense** command to verify the PIM route notification method in use. For example:

```
-> show ip pim sparse
Status                      = enabled,
Keepalive Period            = 210,
Max RPs                     = 32,
Probe Time                  = 5,
Register Checksum           = header,
Register Suppress Timeout   = 60,
RP Threshold                = 1,
SPT Status                  = enabled,
BIDIR Status                = disabled,
BIDIR Periodic Interval     = 60,
BIDIR DF Abort Status       = disabled,
BFD Status                  = disabled,
MoFRR Status                = disabled,
MoFRR All Routes Status     = disabled,
MBR All Sources Status      = enabled,
MBR Operational Status      = enabled

-> show ip pim dense
Status                        = enabled,
Source Lifetime               = 210,
State Refresh Interval        = 60,
State Refresh Limit Interval  = 0,
State Refresh TTL             = 16,
BFD Status                    = disabled,
MoFRR Status                  = disabled,
```

```
MBR All Sources Status        = disabled,
MBR Operational Status        = enabled
```

Configuring DVMRP Default Route Advertisement

Advertising a default route on a DVMRP interface on the MBR provides a method for ensuring that sources inside the PIM domain can reach all routers inside the DVMRP domain. To enable default route advertisement for a specific DVMRP interface, use the **ip dvmrp interface mbr-default-information** command. For example:

```
-> ip dvmrp interface mbr-default-information enable
```

When enabling this type of advertisement, make sure that the default route is not advertised on the MBONE.

To disable the default route advertisement, use the ip dvmrp interface mbr-default-information command with the disable option. For example:

```
-> ip dvmrp interface mbr-default-information disable
```

Use the **show ip dvmrp interface** command to verify whether or not a DVMRP is advertising a default route. For example:

```
-> show ip dvmrp interface

Total 4 Interfaces

Interface Name   Vlan  Metric  Admin-Status  Oper-Status  BFD-Status MBR-Default
-----------------+-----+------+-------------+-------------+----------+-----------
vlan-4              4       1   Disabled      Disabled      Disabled   Disabled
vlan-6              6       1    Enabled       Enabled      Disabled    Enabled
```

CLI Configuration Example

As previously described in this chapter, configuring an OmniSwitch to operate as a multicast border router (MBR) requires configuring PIM and DVMRP first then enabling MBR functionality. This section provides three sample configurations for configuring an OmniSwitch MBR, including commands used for PIM and DVMRP.

Example 1: Default MBR Protocol Configuration

In this example, the switch is configured to act as an MBR but with default MBR settings. PIM will only notify DVMRP about the routes to directly connected subnets and not the routes to all sources. DVMRP will not advertise the default route on any of the DVMRP interfaces.

```
-> ip load pim
-> ip pim interface "vlan-2"
-> ip pim interface "vlan-3"
-> ip pim dense group 225.0.0.0/24
-> ip pim dense admin-state enable
-> ip load dvmrp
-> ip dvmrp interface "vlan-4"
-> ip dvmrp interface "vlan-6"
-> ip dvmrp admin-state enable
-> ip mroute mbr admin-state enable
```

Example 2: DVMRP Default Route

In this example, the switch is configured to act as an MBR but also enables DVMRP to advertise the default route on "vlan-6", but not on "vlan-4".

```
-> ip load pim
-> ip pim interface "vlan-2"
-> ip pim interface "vlan-3"
-> ip pim dense group 225.0.0.0/24
-> ip pim dense admin-state enable
-> ip load dvmrp
-> ip dvmrp interface "vlan-4"
-> ip dvmrp interface "vlan-6"
-> ip dvmrp interface "vlan-6" mbr-default-information enable
-> ip dvmrp admin-state enable
-> ip mroute mbr admin-state enable
```

Example 3: PIM Routes to All Sources

In this example, the switch is configured to act as an MBR but also enables PIM to notify DVMRP about the routes to all sources.

```
-> ip load pim
-> ip pim interface "vlan-2"
-> ip pim interface "vlan-3"
-> ip pim dense group 225.0.0.0/24
-> ip pim mbr all-sources
-> ip pim dense admin-state enable
-> ip load dvmrp
-> ip dvmrp interface "vlan-4"
-> ip dvmrp interface "vlan-6"
-> ip dvmrp admin-state enable

-> ip mroute mbr admin-state enable
```

Verifying the MBR Configuration

A summary of the **show** commands used for verifying the OmniSwitch MBR configuration is given here:

show ip mroute mbr	Displays the MBR administrative status and the MBR protocol registration status for PIM and DVMRP.
show ip pim sparse **show ip pim dense**	Displays the global parameter configuration for PIM-SM or PIM-DM, including the PIM operational status for MBR and the PIM route notification status.
show ip dvmrp	Displays the global parameter configuration for DVMRP, including the DVMRP operational status for MBR.
show ip dvmrp interface	Displays the DVMRP interface configuration, which includes the MBR default route advertisement status for each interface.

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

A Software License and Copyright Statements

This appendix contains ALE USA, Inc. and third-party software vendor license and copyright statements.

ALE USA, Inc. License Agreement

ALE USA, INC. SOFTWARE LICENSE AGREEMENT

IMPORTANT. Please read the terms and conditions of this license agreement carefully before opening this package.

By opening this package, you accept and agree to the terms of this license agreement. If you are not willing to be bound by the terms of this license agreement, do not open this package. Please promptly return the product and any materials in unopened form to the place where you obtained it for a full refund.

1. **License Grant.** This is a license, not a sales agreement, between you (the "Licensee") and ALE USA, Inc. ALE USA, Inc. hereby grants to Licensee, and Licensee accepts, a non-exclusive license to use program media and computer software contained therein (the "Licensed Files") and the accompanying user documentation (collectively the "Licensed Materials"), only as authorized in this License Agreement. Licensee, subject to the terms of this License Agreement, may use one copy of the Licensed Files on the Licensee's system. Licensee agrees not to assign, sublicense, transfer, pledge, lease, rent, or share their rights under this License Agreement. Licensee may retain the program media for backup purposes with retention of the copyright and other proprietary notices. Except as authorized under this paragraph, no copies of the Licensed Materials or any portions thereof may be made by Licensee and Licensee shall not modify, decompile, disassemble, reverse engineer, or otherwise attempt to derive the Source Code. Licensee is also advised that ALE USA, Inc. products contain embedded software known as firmware which resides in silicon. Licensee may not copy the firmware or transfer the firmware to another medium.

2. **ALE USA, Inc.'s Rights.** Licensee acknowledges and agrees that the Licensed Materials are the sole property of ALE USA, Inc. and its licensors (herein "its licensors"), protected by U.S. copyright law, trademark law, and are licensed on a right to use basis. Licensee further acknowledges and agrees that all rights, title, and interest in and to the Licensed Materials are and shall remain with ALE USA, Inc. and its licensors and that no such right, license, or interest shall be asserted with respect to such copyrights and trademarks. This License Agreement does not convey to Licensee an interest in or to the Licensed Materials, but only a limited right to use revocable in accordance with the terms of this License Agreement.

3. **Confidentiality.** ALE USA, Inc. considers the Licensed Files to contain valuable trade secrets of ALE USA, Inc., the unauthorized disclosure of which could cause irreparable harm to ALE USA, Inc. Except as expressly set forth herein, Licensee agrees to use reasonable efforts not to disclose the Licensed Files to any third party and not to use the Licensed Files other than for the purpose authorized by this License Agreement. This confidentiality obligation shall continue after any termination of this License Agreement.

4. **Indemnity.** Licensee agrees to indemnify, defend and hold ALE USA, Inc. harmless from any claim, lawsuit, legal proceeding, settlement or judgment (including without limitation ALE USA, Inc.'s reasonable United States and local attorneys' and expert witnesses' fees and costs) arising out of or in connection with the unauthorized copying, marketing, performance or distribution of the Licensed Files.

5. **Limited Warranty.** ALE USA, Inc. warrants, for Licensee's benefit alone, that the program media shall, for a period of ninety (90) days from the date of commencement of this License Agreement (referred to as the Warranty Period), be free from defects in material and workmanship. ALE USA, Inc. further warrants, for Licensee benefit alone, that during the Warranty Period the Licensed Files shall operate substantially in accordance with the functional specifications in the User Guide. If during the Warranty Period, a defect in the Licensed Files appears, Licensee may return the Licensed Files to ALE USA, Inc. for either replacement or, if so elected by ALE USA, Inc., refund of amounts paid by Licensee under this License Agreement. EXCEPT FOR THE WARRANTIES SET FORTH ABOVE, THE LICENSED MATERIALS ARE LICENSED "AS IS" AND ALE USA, INC. AND ITS LICENSORS DISCLAIM ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING (WITHOUT LIMITATION) ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO LICENSEE. THIS WARRANTY GIVES THE LICENSEE SPECIFIC LEGAL RIGHTS. LICENSEE MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

6. **Limitation of Liability.** ALE USA, Inc.'s cumulative liability to Licensee or any other party for any loss or damages resulting from any claims, demands, or actions arising out of or relating to this License Agreement shall not exceed the license fee paid to ALE USA, Inc. for the Licensed Materials. IN NO EVENT SHALL ALE USA, INC. BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, OR EXEMPLARY DAMAGES OR LOST PROFITS, EVEN IF ALE USA, Inc. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION TO INCIDENTAL OR CONSEQUENTIAL DAMAGES MAY NOT APPLY TO LICENSEE.

7. **Export Control.** This product is subject to the jurisdiction of the United States. Licensee may not export or reexport the Licensed Files, without complying with all United States export laws and regulations, including but not limited to (i) obtaining prior authorization from the U.S. Department of Commerce if a validated export license is required, and (ii) obtaining "written assurances" from licensees, if required.

8. **Support and Maintenance.** Except as may be provided in a separate agreement between ALE USA, Inc. and Licensee, if any, ALE USA, Inc. is under no obligation to maintain or support the copies of the Licensed Files made and distributed hereunder and ALE USA, Inc. has no obligation to furnish Licensee with any further assistance, documentation or information of any nature or kind.

9. **Term.** This License Agreement is effective upon Licensee opening this package and shall continue until terminated. Licensee may terminate this License Agreement at any time by returning the Licensed Materials and all copies thereof and extracts therefrom to ALE USA, Inc. and certifying to ALE USA, Inc. in writing that all Licensed Materials and all copies thereof and extracts therefrom have been returned or erased by the memory of Licensee's computer or made non-readable. ALE USA, Inc. may terminate this License Agreement upon the breach by Licensee of any term hereof. Upon such termination by

ALE USA, Inc., Licensee agrees to return to ALE USA, Inc. or destroy the Licensed Materials and all copies and portions thereof.

10. **Governing Law.** This License Agreement shall be construed and governed in accordance with the laws of the State of California.

11. **Severability.** Should any term of this License Agreement be declared void or unenforceable by any court of competent jurisdiction, such declaration shall have no effect on the remaining terms herein.

12. **No Waiver.** The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

13. **Notes to United States Government Users.** Software and documentation are provided with restricted rights. Use, duplication or disclosure by the government is subject to (i) restrictions set forth in GSA ADP Schedule Contract with ALE USA, Inc.'s reseller(s), or (ii) restrictions set forth in subparagraph (c) (1) and (2) of 48 CFR 52.227-19, as applicable.

14.**Third Party Materials.** Licensee is notified that the Licensed Files contain third party software and materials licensed to ALE USA, Inc. by certain third party licensors. Some third party licensors are third part beneficiaries to this License Agreement with full rights of enforcement. Please refer to the section entitled "Third Party Licenses and Notices" on page -4 for the third party license and notice terms.

Third Party Licenses and Notices

Legal Notices applicable to any software distributed alone or in connection with the product to which this document pertains, are contained in files within the software itself located at: **/flash/foss**.

Also, if needed, we provide all FOSS (Free and Open Source Software) source code used in this release at the following URL: https://github.com/Alcatel-LucentEnterpriseData.

Index

E

I

M

N

O

P

R

S

V